2.2.1 - The institution assesses the learning levels of the students and organizes special Programmes for advanced learners and slow learners

S.No.	Document Attached	Page Number
1.	Induction Program Time Table (B.Tech Ist Year, MCA, MBA, B.Pharma)	2
2.	3 G Policy for Gradual, Growing and Gallant Learner	8
3.	Incentive Policies (University Rank Holders, Branch Toppers & GATE/GPAT qualified)	11
4.	Higher Education Preparation Student Clubs (GATE & GPAT	15
5.	Skill Development Programmes (Soft Skill, Skill Development Finishing School, CCP Cell)	53
6.	Student Manual	https://www.kiet.edu/student-handbook

B.Tech First Year

KIET Group of Institutions, Delhi-NCR, Ghaziabad

Department of B. Tech. I Year and Applied Sciences

Time Table of Induction Program, B. Tech. I Year (2022-23), Nov. 01 - 26, 2022

Date/ Day	06:30 am-07:30 am	First Half		12:30 pm-01:30 pm	Second Half		05:30 pm-07:00 pm
01/11/2022 Tuesday	Yoga Activities	Reporting of Students/ Parents Interactive Session			Literary Activities/ Universal Human Values		Sports Activities
02/11/2022 Wednesday	Yoga Activities	Reporting of Students/ Parents Interactive Session Literary Activities/ Universal Human Values	International Education Fai (Interactive Knowledge Exchange Program)		International Education Fair (Interaction Knowledge Exchange Program)	Literary Activities/ Universal Human Values	Sports Activities
03/11/2022 Thursday	Yoga Activities	Reporting of Students/ Parents Interactive Session Creative Arts/ Literary Activities/ Cultural Activities/ Universal Human Values			Creative Artss/ Literary Activities/ Cultural Activities/ Universal Human Values		Sports Activities
04/11/2022 Friday	Yoga Activities	Reporting of Students/ Parents Interactive Session Creative Arts/ Literary Activities/ Cultural Activities/ Universal Human Values			Creative Artss/ Literary Activities/ Cultural Activities/ Universal Human Values		Sports Activities
05/11/2022 Saturday	Yoga Activities	Sports Activities/ Literary Activities			Sports Activities/ Literary Activities		Sports Activities
06/11/2022 Sunday	Yoga Activities	Sports Activities/ Literary Activities			Sports Activities/ Literary Activities		Sports Activities
07/11/2022 Monday	Yoga Activities	Presentation by Students of Extra-Curricular Activities Club			Presentation by Students of Extra-Curricular Activities Club	Session on Emotional Wellness	Sports Activities
08/11/2022 Tuesday	Yoga Activities	Creative Arts/ Literary Activities/ Cultural Activities/ Universal Human Values		*LUNCH*	Creative Arts/ Literary Activities/ Cultural Activities/ Universal Human Values		Sports Activities
09/11/2022 Wednesday	Yoga Activities	Parent's Interactive Session/ Reporting of Students			Kathak Dance Event By SPIC MACAY		Sports Activities
10/11/2022 Thursday	Yoga Activities	Parent's Interactive Session/ Reporting of Students Visit to Engineering Departments (ME, CE, EEE, ECE, CSE)			Creative Arts/ Literary Activities/ Cultural Activities/ Universal Human Values		Sports Activities
11/11/2022 Friday	Yoga Activities	Parent's Interactive Session/ Reporting of Students Creative Arts/ Literary Activities/ Cultural Activities/ Universal Human Values			Creative Arts/ Literary Activities/ Cultural Activities/ Universal Human Values		Sports Activities
12/11/2022 Saturday	Yoga Activities	Creative Arts/ Literary Activities/ Cultural Activities/ Universal Human Values			Visit to Engineering Departments (CS, CSIT, IT, CSAI, CSML)		Sports Activities
13/11/2022 Sunday	Yoga Activities	Sports Activities/ Literary Activities			Sports Activities/ Literary Activities		Sports Activities
14/11/2022 Monday	Yoga Activities	Creative Arts/ Universal Human Values	Motivational Session on Entrepreneurship by Shri Sushil Aggarwal		Webinar on "Ek Prayas to Save the Youth Against Tobacco Outrage" organized by AKTU and hosted by KIET Group of Institutions, Delhi-NCR, Ghaziabad		Sports Activities

Dr CM Batra

Dr S.K. Tiwary

Dr. Shailendra Kumar Tiwary
Dean
B. Tech 1 Year & Applied Sciences

KIET Group of Institutions, Delhi-NCR, Ghaziabad
Department of B. Tech. I Year and Applied Sciences
Time Table of Induction Program, B. Tech. I Year (2022-23), Nov. 01 - 26, 2022

Date/ Day	06:30 am-07:30 am	First Half			12:30 pm-01:30 pm	Second Half		05:30 pm-07:00 pm
15/11/2022 Tuesday	Yoga Activities	Creative Arts/ Universal Human Values		Motivational Session by Officials of AICTE		Address Session by Dean (Innovation & Entrepreneurship) & Presentation by Start up Companies at TBI-KIET		Sports Activities
16/11/2022 Wednesday	Yoga Activities	Creative Arts/ Universal Human Values		Motivational Session on National Cadet Corps (NCC) by Brig. Salabh Sonal		Innovative Project Exhibition by B. Tech. I Year Students		Sports Activities
17/11/2022 Thursday	Yoga Activities	Literary Activities/ Universal Human Values		Interactive Session on Brain Mapping by Mr. Vikas Nagru		Motivational Session by IAS Officers	Literary activities/ Universal Human Values	Sports Activities
18/11/2022 Friday	Yoga Activities	Literary Activities/ Universal Human Values		Motivational Session by Officials of DST		Visit to Annual Technical Fest (INNOTECH) at KIET Group of Institutions	Literary activities/ Universal Human Values	Sports Activities
19/11/2022 Saturday	Yoga Activities	Sports Activities/ Literary Activities				Sports Activities/ Literary Activities		Sports Activities
20/11/2022 Sunday	Yoga Activities	Visit to Local Area				Sports Activities/ Literary Activities		Sports Activities
21/11/2022 Monday	Yoga Activities	Proficiency Module (Maths)	Proficiency Module (Chemistry)	Creative Arts	L U N C H	Address by Dean (B. Tech. I Year & Applied Sciences)	Creative Arts	Sports Activities
22/11/2022 Tuesday	Yoga Activities	Proficiency Module (Physics)	Proficiency Module (Electrical)	Creative Arts		Address by HODs (B. Tech. I Year)	Creative Arts	Sports Activities
23/11/2022 Wednesday	Yoga Activities	Proficiency Module (Electronics)	Proficiency Module (Mechanics)	Creative Arts		Address by Director & Joint Director	Creative Arts	Sports Activities
24/11/2022 Thursday	Yoga Activities	Proficiency Module (Soft Skills)	Proficiency Module (Computer Prog.)	Creative Arts		Address by Dean (Academics)	Creative Arts	Sports Activities
25/11/2022 Firday	Yoga Activities	Proficiency Module (Maths)	Proficiency Module (Soft Skills)	Creative Arts		Address by Dean (SW)	Creative Arts	Sports Activities
26/11/2022 Saturday	Yoga Activities	Closing Ceremony Celebration				Closing Ceremony Celebration		

Dr. C. M. Batra
SIP Coordinator &
Asscociate Dean (B. Tech. I year & Applied Sciences)

Dr. Shailendra Kumar Tiwary
Dean (B. Tech. I year & Applied Sciences)

Dr. Shailendra Kumar Tiwary
Dean
B. Tech 1 Year & Applied Sciences

Dr. Manoj Goel
Joint Director

KIET Group of Institutions, Delhi-NCR, Ghaziabad
Department of Computer Applications (NBA Accredited)

(An ISO – 9001: 2015 Certified & 'A+' Grade accredited Institution by NAAC)

Orientation Program for MCA Batch 2022-24

DAY/ HOUR	10:00 AM – 10:50 AM	10:50 AM – 11:40 AM	11:40 AM – 12:30 PM	12:30 PM – 1:30 PM	01:30 PM – 02:20 PM	02:20 PM – 03:10 PM	03:10 PM – 04:00 PM
07.11.2022 (Monday)	Snack/ Food Distribution **@MCA Office**	Inaugural Address By **Head of Department**	Academic Activities By **Dean AC**	L	Student Welfare Activities By **Dean SW**	Importance of Internal Complain Cell By **Head-ICC**	Session on Significance of Human Values By **Dr. Shashank**
09.11.2022 (Wednesday)	Departmental Placement Cell (DPC) By **Mr. Ankit**	Departmental Projects By **Dr. Sangeeta**	UP Govt. Schemes & Scholarships for Education By **Mr. Deepak (Admin Dept.)**	U	International Opportunities & Foreign Languages By **Head, PR&IR**	Motivational Talk By **Head, CRPC**	Substantiveness of Competitive Coding (CCIP) By **Mr. Prashant**
10.11.2022 (Thursday)	Skill Development & Finishing School By **Head, SDFS**	Professional Communication By **Dr. Sonia Gouri**	Innovation & Entrepreneurship Cell By **Dr. KLA Khan**	N	Academic Practices By **Addl. Head**	Importance of Aptitude skills By **Mr. Vinod Ag.**	Role of Technical Business Incubator (TBI) By **Mr. Saurav**
11.11.2022 (Friday)	Departmental Student Development Cell (DSDC) By **Team DSDC**	Usage of Departmental Moodle By **Dr. Amit Kumar**	Role of Soft Skills for MCA Grads By **Mr. Himanshu & Ms. Arunita**	C H	Briefing KIET ERP By **Head, ITSS**	Departmental Outcome Based Education Practices By **Team DOC**	
12.11.2022 (Saturday)	**REGISTRATION**						

Dr. Arun Kr. Tripathi
(Head – CA)

KIET Group of Institutions, Delhi-NCR, Ghaziabad

Department of Computer Applications (NBA Accredited)

(An ISO – 9001: 2015 Certified & 'A+' Grade accredited Institution by NAAC)

Orientation Program for MCA Batch 2022-24

DAY/ HOUR	10:00 AM – 10:50 AM	10:50 AM – 11:40 AM	11:40 AM – 12:30 PM	12:30 PM – 1:30 PM	01:30 PM – 02:20 PM	02:20 PM – 03:10 PM	03:10 PM – 04:00 PM
07.11.2022 (Monday)	Snack/Food Distribution **@MCA Office**	Inaugural Address By **Head of Department**	Academic Activities By **Dean AC**	**LUNCH**	Student Welfare Activities By **Dean SW**	Importance of Internal Complain Cell by **Head-ICC**	Session on Significance of Human Values By **Dr. Shashank**
09.11.2022 (Wednesday)	Skill Development & Finishing School By **Head, SDFS**	Professional Communication By **Dr. Sonia Gouri**	Usage of Departmental Moodle By **Dr. Amit Kumar**		Motivational Talk By **Head, CRPC**	Innovation & Entrepreneurship Cell By **Dr. KLA Khan**	Importance of Aptitude skills By **Mr. Sarvendra**
10.11.2022 (Thursday)	Role of Soft Skills for MCA Grads By **Mr. Himanshu & Ms. Arunita**	Departmental Projects By **Dr. Sangeeta**	UP Govt. Schemes & Scholarships for Education By **Mr. Deepak (Admin Dept.)**		International Opportunities & Foreign Languages By **Head, PR&IR**	Academic Practices By **Addl. Head**	Departmental Placement Cell **(DPC)** By **Mr. Ankit Verma**
11.11.2022 (Friday)	Departmental Outcome Based Education Practices By **Team DOC**		Substantiveness of Competitive Coding (CCIP) By **Mr. Prashant**		Departmental Student Development Cell (DSDC) By **Team DSDC**	Briefing KIET ERP By **Head, ITSS**	Role of Technical Business Incubator (TBI) By **Mr. Saurav**
12.11.2022 (Saturday)	**REGISTRATION**						

Dr. Arun Kr. Tripathi
(Head – CA)

KIET GROUP OF INSTITUTIONS GHAZIABAD
KIET SCHOOL OF MANAGEMENT

Orientation Programme
MBA-1st Year (Batch 2022-24)

Details of Programme

Day-1 (07/11/2022) Venue-MBA 3rd Floor			
1.	10:30 – 11:30	Brief about Institute & Department by HOD	Dr Shivani Agarwal
2.	11:30 – 12:30	Case Analysis	Dr Binkey Srivastava
3.	12:30-1:30	**Lunch**	
4.	1:30 – 2:30	Class Coordinators' Session	First Year Coordinators
5.	2:30-3:30	CLUB Coordinators' Session	Dr Arunima Mishra; Dr Deepa, Dr Sudheer Kumar, Dr Prateek Gupta
Day-2 (08/11/2022)			
HOLIDAY- Guru Nanak Jayanti			
Day-3 (09/11/2022) Venue- CRPC Hall			
1.	9:00 – 10:00	Registration- Kit Distribution	First Year Coordinators
2.	10:00- 10: 10	Address by Joint Director	Dr. Shivani Agarwal
3.	10:10 –10:20	Address by Dean (Academics)	
4.	10:20 – 10:30	Brief about Institute & Department by HOD	
5.	10:30-11:00	Session by Head-Alumni	
6.	11:00 – 12:30	International Affairs	Dr. Preeti Chitkara
7.	12:30-1:30	**Lunch**	
8.	1:30 – 2:30	Guest Lecture	Dr. Ranchay Bhateja
9.	2:30-3:30	Interaction with Dean SW	Dr. Shivani Agarwal
10.	3:30 –3:45	POSH	Mr. Anjan Kumar
	3:45 – 4:15	EAP/Ashish Thombre	Dr. Prateek Gupta
	4:15-4:30	Kavyanjali Club	
Day-4 (10/11/2022)			
1.	9:30AM-4:30PM	Industrial Visits	Faculty as per Availability
Day-5 (11/11/2022)			
1.	9:00-4:30	International Conference	All Faculty Members
Day-6 (12/11/2022)			
1.	9:00-4:30	International Conference	All Faculty Members

HOD-KSOM

KIET Group of Institutions, Delhi-NCR, Ghaziabad
KIET School of Pharmacy (NBA Accredited)

(An ISO – 9001: 2015 Certified & 'A+' Grade accredited Institution by NAAC)

Orientation Program for B. Pharm Batch 2022-24

DAY/ HOUR	10:00 AM – 10:50 AM	10:50 AM – 11:40 AM	11:40 AM – 12:30 PM	12:30 PM – 1:30 PM	01:30 PM – 02:20 PM	02:20 PM – 03:10 PM	03:10 PM – 04:00 PM
07.11.2022 (Monday)	Inaugural Address By **Prof. (Dr.) K. Nagarajan-** Principal-KSOP	Know Your Faculty - By **Mr. Praveen K. Dixit**	PCI Curriculum/Examinations Policy **Prof. (Dr.) K. Nagarajan/ Dr.Parul Grover**	**L U N C H**	Introduction Session by Students **(Class Coordinators)**	Importance of Internal Complaint Cell and Discipline By **Dr. Roma/Dr. Abhay**	Session on Significance of Human Values By **Dr. Daksh Bhatia**
09.11.2022 (Wednesday)	Departmental Placement Cell (DPC) By **Ms. Vidhu Saxena**	Institutional Academic Activities By **Prof.(Dr.)Anil K. Ahlawat** Dean AC	Student Welfare Activities By **Prof.(Dr.) Satish Kumar** Dean SW		Academic Practices/Projects/ By **Prof. (Dr.) NG Rao** -Addl. Head	UP Govt. Schemes & Scholarships for Education **By Mr. Deepak** (Admin Dept.)	Importance of Aptitude skills By **Dr.Meetu Kumar ,ASS**
10.11.2022 (Thursday)	Role of Technical Business Incubator (TBI) **By Mr. Saurav**	Address By **Mr. Anup Srivastava**, Registrar, KIET	Usage of Departmental Moodle By **Mr.Surya Prakash**		International Opportunities & Foreign Languages By **Dr. Preeti Chitkara -Head, PR&IR**	Fundamentals of IPR and Patent Search By **Dr. Richa Goel**	Departmental Student Development Cell (DSDC) By **Mr.Praveen K.Dixit**
11.11.2022 (Friday)	Skill Development & Finishing School By **Dr. Ajay K Srivastava** Head, SDFS	Session on Ethics and Values By **Mr.Praveen K.Dixit**	Usage of KIET ERP Module By **Mr.Vinay Ahlawat/Mr.Sanjeev Chauhan**		Research Accolades By **Prof.(Dr.)Ashu Mittal**	Departmental Outcome Based Education Practices By Team DOC	
12.11.2022 (Saturday)	Importance of Cyber Security by **Prof. (Dr.) Arun K. Tripathi-Head-MCA 10am-11 am** **Interaction with Parents/Students 11:00am-12:00am**						

Prof.(Dr.)K.Nagarajan
Principal,KSOP

Mr.Praveen K.Dixit/Mr.Pankaj Bhatt
Orientation-2022 Coordinator(s)

KIET Policy for Gradual/Growing /Gallant Learners

Date: 17th Nov' 2022

In KIET, 3 Basket System was introduced initially in July 2003 and modified in July 2018 with the following criteria:

- Basket 3: Students having Marks less than 40%
- Basket 2: Students having marks between 40% and 60%
- Basket 1: Students having marks greater than 60%

In the year 2020, a KIET policy was initiated in which students are bifurcated as Slow/Average/Advance Learners.

Every individual is exclusive with completely different I.Q. levels. Some students have the potential of grabbing quickly. While, some students may have moderate learning and take time to grab the things. It is observed that the word slow learner is somehow impacting the students psychologically.

In this regard, few internal meetings were conducted at institute level after that a Google form for feedback and suggestions with fifteen different combinations for each category was circulated among the students and faculty members. Our institute always welcomes innovative ideas to enhance the system for the betterment of the students. As learners, students play a crucial and active role in the institute.

We are glad to announce the 3G learners (Gradual, Growing, Gallant) as new names in place of Slow/Average/Advance Learners which is effective from 14th Nov'2022 on the very auspicious occasion of Children's Day.

This policy is modified with the following criteria:

- **Gradual**: Students having Marks less than 40%
- **Growing**: Students having marks between 40% and 60%
- **Gallant**: Students having marks greater than 60%

1. Gradual Learners (Students having Marks less than 40%)

- The performance of all the gradual learners is the responsibility of the respective teachers teaching in particular section. For the same, subject teacher may call the students personally or in group for teaching/ guiding/counselling. This will enable to attain Zero PCP Policy by improving the performance.

- Departments will identify gradual learner students on the basis of odd/even semester university examination result and update the list based on their performance in Pre-CT, CTs and PUE. Departments shall plan remedial classes (4 lectures/week viz. preferably 7th and 8th lecture), providing extra support/guidance/solution of last year question papers etc. to improve their academic performance.

- **Continuous assessment of gradual learners is required:** For which all subject faculty members may give the second set of question paper of CTs/PUE to all the gradual learners and ask them to solve it and discuss it with subject faculty member in remedial classes.

- **The Weightage of above tests will be included in PIT marks as follows:**
  PIT Marks = Average of marks secured in the test given in remedial classes (second set of question paper of CTs and PUE).

- The primary focus of departments for gradual learners must be the presence of these learners in the regular/ remedial classes scheduled by the department.

- Coding Classes for basic concept understanding can be organized.

2. Growing Learners (Students having marks between 40% and 60%)

- Soft Skill Classes/ Competitive Coding Classes/ Certification of various MOOC courses can be planned for skill development and improving overall academic performance.

- Participation in Club Activities (Dev Up Club/Industrial Electronics and Control Club/ ME Design Club/The Quiz club of KIET/GPAT student's club/E-Yantra/ Marketing Club/ Electric Vehicle Club/ The Material and Metallurgy Testing Club/ Coding Clubs etc.) may be planned.

- Mentors can provide good text books other than reference books written in subject syllabus for better understanding of the subject. This helps in solving analysis problems related to that subject.

- The primary focus of departments for growing learners must be the presence of these learners in the regular classes scheduled by the department. Also, to motivate these learners to participate more in different clubs activities to boost their overall performance.

3. Gallant Learners (Students having marks greater than 60%)

- Special Classes (GATE/GPAT/GRE/CAT/MAT)/ Participation in Coding Competitions (Hackathon/HackerEarth/HackerRank/GitHub)/ Certification of various advance MOOC courses.

- Participation in Club Activities (SAE India Collegiate Club of KIET/INNOGEEKS/Salesforce Learning@KIET/TBI Incubation lab cum E-Cell/Dinobots/

Pharma Innovation Club/Instronix/SYS-MATIC automation club/DevUp Club/ DSDL Club/ Google DSC/ Coding Clubs/ Socio Tech Innovation Club, KODERS KORNER etc.)

- Research Activities including Project Grant/Research Paper Publication/Patent Publication/Working on any real time project may be planned.

- Mentors can provide good text books other than reference books written in subject syllabus to advance learners for better understanding of the subject. This helps in solving analysis problems related to that subject.

- The primary focus of departments for gallant learners must be the presence of these learners in the regular classes scheduled by the department. Also, to motivate these learners to participate more in different clubs activities and coding competitions to enhance their overall performance.

- The practical list of the gallant learners should be prepared separately in which problem should match with Industry standards.

- A separate set of question papers for CTs/PUE should be prepared for gallant learners and will be distributed in examination as per departmental gallant learners list.

Note:

1. Only students having attendance above 60% are eligible to participate in different club activities, research activities and coding competitions as Growing or Gallant Learner.

2. All the students will create their digital profile using Git hub repository, Hacker rank/Code chef and Leet code star ratings.

23/11/2022

Dean Academics

Director

23 NOV 22

KIET GROUP OF INSTITUTIONS

DIRECTOR OFFICE <directoroffice@kiet.edu>

Students (For Information) - Institutes Policy for rewarding Top -10 University Rank Holders

6 messages

DIRECTOR OFFICE <directoroffice@kiet.edu> Fri, May 24, 2019 at 2:04 PM
To: students_kiet@kiet.edu
Cc: "KIET Group (All)" <kietians@kiet.edu>

Dear Students,

You must be aware that Institute has well defined policy to **felicitate & reward** ***Top 10 University Rank Holders*** during **annual Convocation function**.

We are pleased to share Institutes **existing policy** i.e. Reward Scheme for Top-10 University Rank Holders **(Revised in Mar'17)** for your kind information.

- **University Gold Medalist** - **Rs. 50000/- each**
- **University Silver Medalist** - **Rs. 40,000/- each**
- **University Bronze Medalist** - **Rs. 30,000/- each**
- **University 4th -10th Rank Holders** - **Rs. 20,000/- each**

We look forward to have more number of University Rank Holders every year.

Wishing you All the Best!!

Thanks & regards

Director Office

KIET Group of Institutions

Delhi-NCR, Meerut Road (NH-58)

Ghaziabad, U.P. PIN - 201206

directoroffice@kiet.edu | http://www.kiet.edu/

KIET Vision: To achieve *excellence in technical education* and create *competent professionals* for Industry & *Socio-economic development* to meet National and International needs.

DIRECTOR OFFICE <directoroffice@kiet.edu> Thu, Jun 13, 2019 at 9:29 AM
To: DEAN AC <dean_ac@kiet.edu>

Thanks & regards

Ref/ Dir/ 16/ 2020 **Date: 3rd Sep'20**

OFFICE ORDER: 16/2020
Revision – Scholarship/Incentive Scheme (Students)

1. Institute gives recognition to all its students who excel in studies. In view of growing competition & to augment the competitive spirit among students in terms of attaining a new level (academic %) it is decided to **revise the policy w.e.f. Academic Year 2019-20** as per the following details: -

A. Merit based Scholarship for Course / Branch Toppers

1st Position : Rs. 12000/-
2nd Position : Rs. 6000/-

Toppers of the class will be judged on the basis of the **result of both semesters (without carryover/clear pass in first attempt)** and this incentive will be awarded for the following year.

B. Scheme to encourage students to excel in University Examination

Students of **B.Tech** & **B.Pharm 1st to 3rd year**, securing **85%** (***aggregate marks including both semesters***) will be awarded **Rs. 500/-** and for **each % marks above 85%,** will be given **Rs. 200/- extra** in addition to **Rs. 500/-** (e.g. Student who secures 90% will be in awarded Rs.1500/-).

Students of **MCA 1st & 2nd year** and **MBA 1st year** securing **80%** (***aggregate marks including both semesters***) will be awarded **Rs. 500/-** and for **each % marks above 80%,** will be given **Rs. 200/- extra** in addition to **Rs. 500/-** (e.g. Student who secures 85% will be in awarded Rs.1500/-).

However, Branch Toppers at **para 1 A** will **not be eligible** to **get the benefit** under this clause.

2. In addition, a **scheme to encourage undergraduate programme students** with **highest improvement in academic result** in comparison to previous year examination is hereby introduced as per following details:-

A. For B.Tech. 1st Year Students:

(i) **Students admitted with 50% and above PCM at 10+2 level: Top two Students from each branch** securing highest improvement % in First year result in comparison to PCM% at 10+2 level will get incentive as below:

1st highest improvement in % : Rs. 5000/-
2nd highest improvement in % : Rs. 3000/-

(ii) <u>Students admitted with less than 50% PCM at 10+2 level</u>: Top three Students securing highest improvement % of First year result in comparison PCM% at 10+2 level will get incentive as below:

1st highest improvement in % : Rs. 5000/-
2nd highest improvement in % : Rs. 3000/-
3rd highest improvement in % : Rs. 2000/-

B. For B.Tech. 2nd & 3rd Year Students:

Top two Students securing highest improvement % in 2nd year/3rd year result in comparison 1st year / 2nd year result will get incentive as below:

1st highest improvement in % : Rs. 3000/-
2nd highest improvement in % : Rs. 3000/-

C. For B. Pharm 1st Year Students:

Top two Students securing highest improvement % in First year result in comparison to PCB / PCM % at 10+2 level will get incentive as below:

1st highest improvement in % : Rs. 3000/-
2nd highest improvement in % : Rs. 3000/-

D. For B. Pharm 2nd & 3rd Year Students:

Top two Students securing highest improvement % in 2nd year/3rd year result in comparison to 1st year / 2nd year result will get incentive as below:

1st highest improvement in % : Rs. 3000/-
2nd highest improvement in % : Rs. 3000/-

3. The above scheme is applicable **w.e.f Academic Year 2019-20.**

4. With this, **Circular** dated '**16th Mar'15 'Incentive Scheme -Students' stands superseded.**

All the students are advised to do well in studies and avail this opportunity.

Dr. (Col) A Garg
Director
3rd Sep'20

Distribution:

- Joint Director/ Deans/ Principal-KSOP/ HoDs (CSE/IT/EC/EN/EI/CE/ME/AS/MCA/MBA/HS/CS/CSIT), GM-TBI & Dean IEC, Head-CRPC, Head-CAM, Head-AEC, Head-IRCDC, Head-IC, Addl. Head-IIPC, Chairperson-ICC, Head-IT Operations, Manager-IA, Head-HR, Registrar, Admin Officer, Accts Officer, Librarian, Purchase Officer, KIETians

Copy to:

- All Students

Ref/ Dir/10/2021 **Date: 18th Jun'21**

OFFICE ORDER: 10/2021

Incentive Scheme for GATE/GPAT qualified candidates w.e.f. A.Y 2020-21

1. Institute has been recognising/felicitating its **Top 200 (AIR) GATE/GPAT rank holders** with a cash reward/cheque (Rs. 5000/-) during annual convocation as per the approved policy since 2014.

2. In view of increasing importance of GATE/GPAT, Institute has come up with a **new scheme/policy** to facilitate students/ motivate them to crack this National level exams/strive for a good AIR score so that they have a fair chance of getting admission in good Universities for pursuing their higher studies, JRF-CSIR & PSUs for employment purpose etc.

3. The following policy/scheme (as mentioned at category A & B) has been introduced to felicitate GATE/GPAT rank holders w.e.f. **academic year 2020-21**:-

Category A	Category B
Scheme to encourage students to excel in GATE/GPAT examination	**Merit based Scheme for achievers in GATE/GPAT examination**
▪ **B. Tech. 3rd and 4th year students & B. Pharm. final year who qualify** GATE/GPAT exam will be eligible to **receive reimbursement of examination fee** on the basis of All India Rank (AIR) as per below mention subcategories: a. **AIR below 5000** - 100% examination fee b. **AIR below 10000** - 75% examination fee c. **All qualified rank holders (Above AIR 10000)** - 50% examination fee	▪ **B. Tech. 3rd and 4th year students & B. Pharm. final year students** who qualify GATE/GPAT exam with **All India Rank (AIR) less than 1000** will be eligible to receive **special recognition** for his/her performance as per below mention subcategories: a. **AIR 1 to 200 - 100% examination fee + Rs. 5000/-** b. **AIR 201 to 500 - 100% examination fee + Rs. 3000/-** c. **AIR 501 to 1000 - 100% examination fee + Rs. 2000/-**

4. With this, Policy Note (year 2014) mentioned at Para 1 above stands dissolved.

Students to make best use of this opportunity. We expect good number of aspirants qualify GATE/GPAT exams every year.

Dr. (Col) A Garg
Director

Distribution:

- Joint Director/ Deans/ Principal-KSOP/ **HoDs** (CS/IT/CS/CSIT/EC/EN/EI/CE/ME/AS/MCA/MBA/HS), Head-CRPC, Head-CAM, Head-AEC, Head-IC, Head-EC (IEC), Addl. Head-IIPC, Head-IT Operations, Manager-IA, Head-HR, Registrar, Admin Officer, **Accts Officer**, Librarian, Purchase Officer, KIETians

Copy to:

- Students (B.Tech & B.Pharm)
- Chief Coordinator/Coordinator & Team - Higher Education Club (GATE) & GPAT Club

Ref/ Dir/ 14/2020 **11th Jul'20**

OFFICE ORDER: 14/2020

Formation of Student's Club - Graduate Pharmacy Aptitude Test (GPAT) @ KSOP

Reference: Suggestion by students during CRs meet

1. GPAT student's club @ KSOP is hereby constituted with an objective to support the aspiring Pharmacy students (from 3rd semester onwards) towards their preparation for National level exam GPAT.

2. A well-structured policy doc. covering the **Structure of Club** involving faculty & Student's core team their **roles & responsibilities** and **separate guidelines** for each is attached for more detailed understanding and clarity **(Appendix-A)**.

3. **Dr. Puspendra Kumar, KSOP** is hereby assigned the role of **GPAT Club Coordinator** supported by Joint Coordinators, faculty & student's core team under the overall direction and support of Addl. HoD KSOP as mentioned in **Appendix-A**. He shall be performing the above duties in addition to his regular academic engagement.

4. The overall responsibility of running the club's activities smoothly @ dept. level shall be with Principal-KSOP. The entire team associated with **GPAT student's club** at **para 3** shall be responsible to **Principal KSOP** for their duties in their designated capacities.

5. Principal KSOP shall apprise **Dean (A)** on regular basis *w.r.t.* Club's periodic progress/ time table / planning etc.

Dr. (Col) A Garg
Director

Distribution:

- Joint Director/ Deans/ Principal-KSOP/ HoDs (CS/IT/EC/EN/EI/CE/ME/AS/MCA/MBA/HS)/ Addl. HoDs (CSI/CO), GM-TBI & Dean IEC, Head-CRPC, Head-CAM, Head-AEC, Head-IRCDC, Head-IC, Addl. Head-IIPC, Chairperson-ICC, Head-IT Operations, Manager-IA, Head-HR, Registrar, Admin Officer, Accts Officer, Librarian, Purchase Officer

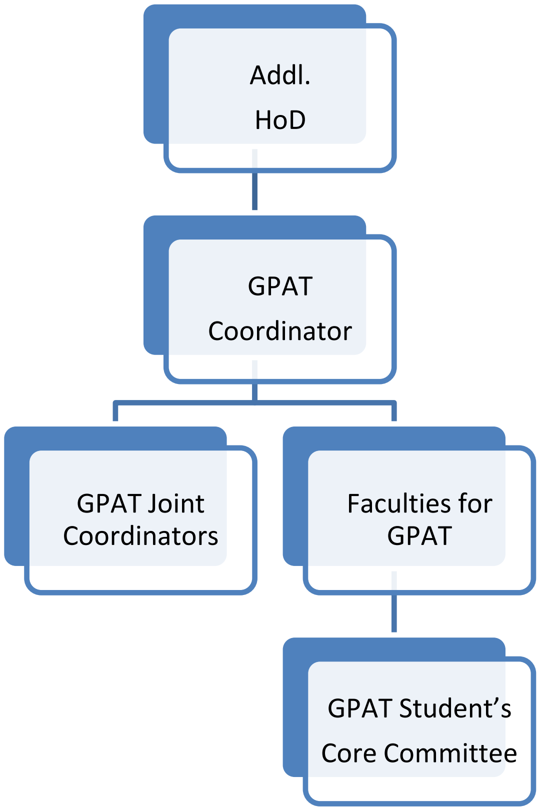
Copy to:
- GPAT Student's Club – All concerned members & faculty
- Dean A, Principal -KSOP

GPAT Student's Club

Introduction

- GPAT (Graduate Pharmacy Aptitude Test) is a national level of exam for Pharmacy students.
- GPAT is conducted by NTA.
- GPAT qualified students are eligible to take admission in reputed institutes with the fellowship of Rs 12400 per month.
- GPAT Scorecard is valid for admission in PhD for many universities.
- GPAT is helpful for NIRF ranking, NBA and NAAC accreditation.
- Students preparing for GPAT will be automatically ready for the future exams and interviews.

Structure of the Team



1. **Addl. HoD:** Dr. K. Nagarajan
2. **GPAT Coordinator:** Dr. Puspendra Kumar
3. **GPAT Joint Coordinator:** Mr. Surya Prakash and Dr. Abhishek Kumar

Roles and Responsibilities

Addl. HoD:

- Responsible to take the decision for the club.
- Review and approve the proposals of coordinator, joint coordinators and faculties.
- Responsible for reporting of monthly activities to Dean Academics for perusal by JD/ Director. **(Annexure-I).**
- To take feedback from Students and Faculty members in every fifteen days, analyse them and to make effective implementation of activities for smooth functioning.
- To take all the decisions about indiscipline students.
- Smooth coordination with Coordinator, joint coordinators, faculties, student's core committee and student members.
- To hold Mock Test, Declaration of Result and Analysis of Result periodically.

Co-ordinator:

- Smooth coordination with Additional HoD, joint coordinators, faculties, student's core committee and student members.
- Responsible for design and smooth conduction of the club.
- Class coordinator for final year students.
- Responsible to communicate the information with the faculties.
- Responsible for taking the MCQs from faculties and conducting the tests.
- Responsible for taking feedback of students on regular basis.
- Responsible to discuss the feedback of students with Additional HoD.

Joint Co-ordinators:

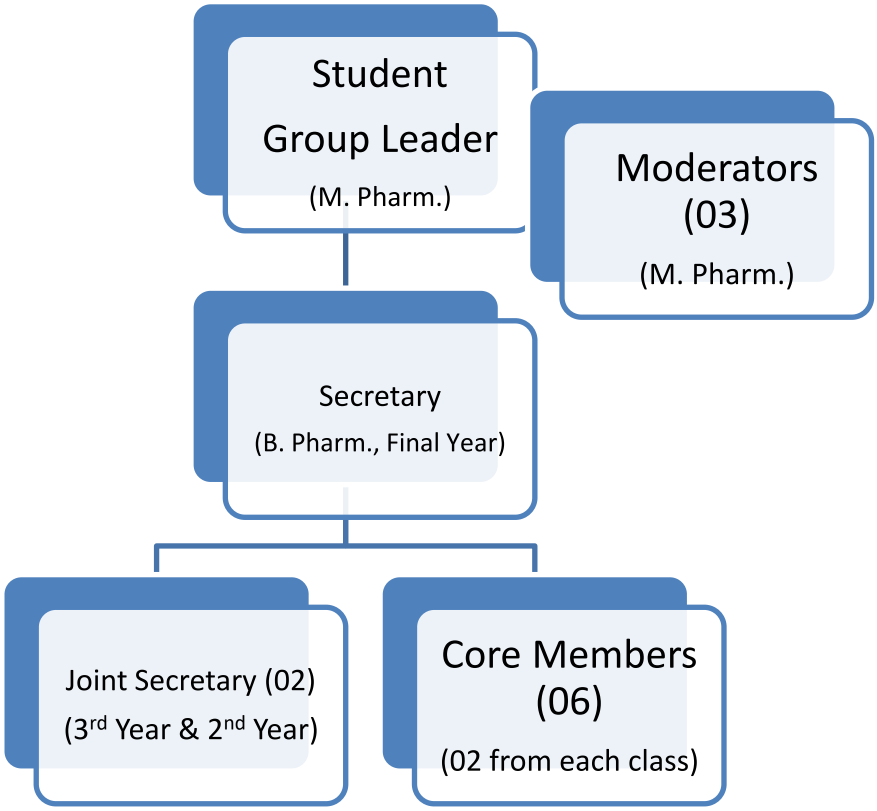
- He / She will complement the coordinator and play the same role in the absence of Coordinator. He will be responsible for design and smooth conduction of the club.
- He will be the class coordinator for final year students.
- He will be responsible to communicate the information with the faculties.
- He will be responsible for taking the MCQs from faculties and conducting the tests.
- He will be responsible for taking feedback of students on regular basis.
- He will be responsible to discuss the feedback of students with Additional HoD.

Faculty members:

They will be responsible for smooth conduction of classes. They will be conducting open discussion among group of students on the covered topics and test questions will be done after each test to improve the understanding of topics and pattern of exam. They will be giving appropriate number of MCQs of the respective subject/topic whenever asked by the co-ordinator/ joint co-ordinator. Faculties will be facilitating the open discussion session to the respective classes.

Appendix-A

Structure of the Student's Core Team



Student Group Leader:	Shubham Sharma (M. Pharm.)
Moderators:	Vardan Gupta (M. Pharm.)
	Shivam Arya (M. Pharm.)
	Varnika Sharma (M. Pharm.)
Secretary:	Dhruv Gaur (B. Pharm., Final Year)
Joints Secretary:	Kavya Gupta (B. Pharm., Third Year)
	Isha Garg (B. Pharm., Second Year)
Members:	Punit Jain (B. Pharm., Final Year)
	Anubha Andhiwal (B. Pharm., Final Year)
	Vartika (B. Pharm., Third Year)
	Km. Anshika Sharma (B. Pharm., Third Year)
	Harshita Sadhana (B. Pharm., Second Year)
	Musaib Ziya (B. Pharm., Second Year)

Role of the Student's Core Committee

Student Group Leader: He / She will lead the student's core committee for various club activities and can take some classes depending upon their interest and availability.

Moderators: They will be the moderator for the student discussion forum, so that the openness of students may be increased and can take some classes depending upon their interest and availability. Open question answers session can be moderated by moderators, it will increase the interactions and concept understanding among the students during and after the classes.

Secretary: He / She is responsible for ensuring that all the affairs of a club are carried out in a smooth and transparent manner. The club secretary has to make necessary arrangements for the club meetings. These include deciding the date, time of the meeting and sending invitations to club members. One of the most important duties of a club secretary is to take down the minutes of a meeting. This includes noting down the important decisions or recommendations that came up in the meeting. The club secretary is also entrusted with keeping a record of the members who attended the meeting and those who were absent.

Joint Secretaries: He / She will complement the secretary and play the same role in the absence of secretary.

Core Committee Members: They will directly communicate with the club members/ students of respective class. Welcomes new members and invites them to club/group activities.

Club Members (Students): Students will attend the classes and they will take part in various activities of the club like discussion forum, questionnaire and doubt sessions etc.

Guidelines for the Students

- Official GPAT classes will start from 8th June, 2020.
- All interested students are requested to register for the club membership and to attend the GPAT Classes **(Annexure-II)**
- Classes will be conducted only through online mode till the regular classes resumed. (5 Hrs/ Week).
- Classes will be conducted online/offline mode after the regular classes resumed. (4 Hrs/ Week).
- The scheduled time table of GPAT Classes should be allotted separately for B. Pharm. (3rd Semester/ 5th Semester/ 7th Semester Students)
- Attendance shall be taken into record to the students attending GPAT classes.
- Students giving the consent to attend GPAT classes to be present in >90% classes. Student fails to achieve the attendance criteria will be debarred from the GPAT classes or have to give genuine reason for not attending.
- If students of 3rd sem. and 5th sem. are not registering, then they will not get the chance to register in next year for 5th and 7th semester GPAT Classes respectively.
- If any student wants to join in later semester, he/she has to score minimum 60% score in three consecutive tests. However, no direct entry in 7th Semester for the students admitted from the session 2018-19.
- If student shall not be able to score 40% or more in any three-consecutive test (with negative marking); then he/she will not be eligible to continue the classes.
- If student shall not be able to score 20% or more in any test (with negative marking); then he/she will not be eligible to continue the classes.
- Online test will be conducted through Moodle and results shall be displayed in WhatsApp group.
- All classes shall be free of cost for all the interested students.
- GPAT guidelines shall be given time to time by faculty members.
- Important topics shall be covered during the classes.
- Feedback will be taken from the students twice in a month.
- Students can discuss the problems with their GPAT Class Coordinators.

Guidelines for the Faculties

- No remuneration shall be given to the faculty members taking GPAT classes.
- Interested Faculty members shall take the classes.
- Class schedule will be given by class coordinators for the respective faculty members. **(Annexure -III)**
- According to the availability of the classes; faculty members will submit the topics to be covered (important topics for GPAT) in advance for further approval process. **(Annexure -III)**
- Addl. HoD will review the topics and approve. Finalized topics will be sent to Dean Academics and Director-KIET.
- Dean Academics, Joint Director and Director (KIET) will monitor the class once in a month.
- Topics to be covered shall be informed to the students at least one week in advance.
- Faculties will submit the report of the class to the respective class coordinator. **(Annexure -IV)**
- Interactive question answer session shall be taken by the faculty members.
- Mentoring and motivation shall be given to the students' time to time.
- Faculty members of respective subject will send at least 10-15 good quality MCQs to conduct the test.
- Students shall be encouraged to solve maximum number of questions every day.
- Previous year questions shall be discussed by the faculty members.
- Coordinator and Joint Coordinators can suggest to change the questions for test.
- GPAT Class Coordinators will conduct the test twice in a month.

List of Faculty Members

S. No.	Name of the Faculty	Specialization	Allotted Subjects to cover
7th Semester			
1.	**Dr. Puspendra Kumar (Class coordinator)**	Analysis and Miscellaneous	Pharmaceutical analysis Pharmacognosy Miscellaneous
2.	Dr. Ashok Jangra	Pharmacology	Pharmacology Microbiology
3.	Dr. Ashu Mittal	Pharmaceutics	Biopharmaceutics and pharmacokinetics Pharmaceutics
4.	Mr. Himanshu Aggarwal	Pharmacology	Pharmacology Microbiology
5.	Dr. K. Nagarajan	Pharm. Chemistry	Biochemistry Pharmaceutical analysis
6.	Dr. Parul Grover	Pharm. Chemistry	Medicinal chemistry Pharmaceutical inorganic chemistry
7.	Mr. Shadab A. Siddiqui	Pharm. Chemistry	Organic chemistry Medicinal chemistry
8.	Ms. Monika Bhardwaj	Pharmacology	Pharmacology Pharmaceutics Biotechnology (plant cell and tissue culture, animal cell culture)
9.	Ms. Richa Goel	Pharmacognosy	Pharmacognosy
10.	Ms. Kiran Sharma	Pharmaceutics	Pharmaceutical jurisprudence Physical pharmacy Pharmaceutics

***Faculties Taking same subjects can distribute the topics with mutual discussion.**

5th Semester			
11.	**Mr. Surya Prakash** **(Class coordinator)**	Pharm. Chemistry	Organic chemistry Biochemistry Miscellaneous
12.	Dr. Abhay Bhardwaj	Pharm. Chemistry	Medicinal chemistry Pharmaceutical analysis
13.	Dr. Deepti Katiyar	Pharmacognosy	Pharmacognosy Pharmaceutical inorganic chemistry
14.	Dr. Mandeep Kumar Arora	Pharmacology	Pharmacology
15.	Ms. Vidhu Saxena	Pharmacology	Pharmacology Pharmaceutical Management
16.	Mr. Anuj Pathak	Pharmaceutics	Pharmaceutics Cosmetics
17.	Mr. Debaprasad Ghosh	Pharmaceutics	Pharmaceutics Pharmaceutical jurisprudence
18.	Ms. Monika Kaurav	Biotechnology	Microbiology Biotechnology

***Faculties Taking same subjects can distribute the topics with mutual discussion.**

3rd Semester			
19.	**Dr Abhishek Kumar (Class coordinator)**	Pharmacology	Pharmacology Miscellaneous
20.	Ms. Lakshmi	Pharmaceutics	Pharmaceutics Dispensing Pharmacy
21.	Mr. Praveen Dixit	Pharmacology	Human anatomy & physiology Pathophysiology
22.	Mr. Sanjeev Chauhan	Pharmaceutics	Dispensing Pharmacy Pharmaceutics
23.	Ms. Garima Kapoor	Pharm. Chemistry	Biochemistry Organic Chemistry
24.	Dr. Roma Ghai	Pharmacology	Human anatomy & physiology Pathophysiology
25.	Ms. Shipra Singhal	Pharm. Chemistry	Organic Chemistry Biochemistry

***Faculties Taking same subjects can distribute the topics with mutual discussion.**

Annexure-I

Class		2nd Year/ 3rd Year/ 4th Year	
S. No.	**Details**	**Planned**	**Conducted**
1.	**Total Number of Classes**		
2.	**Total Number of Tests**		
3.	**Total Number of Open Discussions**		
4.	**Student's Feedback**		
5.	**Faculties remarks**		
6.	**Coordinator/ Joint coordinators remarks**		
7.	**Addl. HoD Remarks**		
8.	**Signature of Addl. HoD**		

Annexure-II

Registration Form

Name of the Student	**:**	
University Roll No.	**:**	
Father's Name	**:**	
Batch	**:**	
Section	**:**	
KIET Mail ID	**:**	
Mobile Number	**:**	
Reason(s) to do the GPAT Classes	**:**	
Declaration	**:**	**I am interested in GPAT Preparation Classes. I will attend the GPAT Classes. I have read the terms and conditions for GPAT Classes and I am giving my consent.**
Signature of the participant	**:**	

Annexure-III

Classes Schedule (Class: 2nd Year)

S. No.	Date of the Lecture/ Test	Assigned Faculty	Topics
1.	08-06-2020	Dr. Abhishek Kumar	
2.	09-06-2020	Ms. Lakshmi	Pharmacy Profession & Introduction to Pharmaceuticals
3.	10-06-2020	Mr. Praveen Dixit	Basic principles of cell injury: Causes, pathogenesis and morphology of cell injury.
4.	11-06-2020	Mr. Sanjeev Chauhan	Weighing methodology
5.	12-06-2020	Ms. Garima Kapoor	Ultrastructure of the cell, functions of various cellular constituents, Applications of biochemical principles to the pharmacy.
6.	15-06-2020	Dr. Roma Ghai	Sense organs
7.	16-06-2020	Ms. Shipra Singhal	A brief review of classification & sources of organic compounds
8.	17-06-2020	Dr. Abhishek Kumar	Pharmacology basics
9.	18-06-2020	Ms. Lakshmi	Definition, importance of pharmaceuticals, areas concerned, scope of Pharmaceutics,
10.	19-06-2020	Mr. Praveen Dixit	Cellular adaptation
11.	20-06-2020 (Test)	Dr. Abhishek Kumar	Test from syllabus covered in last 15 days
12.	22-06-2020	Mr. Sanjeev Chauhan	Handling of prescriptions, labeling instructions for dispensed products
13.	23-06-2020	Ms. Garima Kapoor	Types of carbohydrates, their functions, digestion, & absorption. Aerobic & anaerobic oxidation with energetics. Glycogenesis, glycogenolysis, & gluconeogenesis.
14.	24-06-2020	Dr. Roma Ghai	Infectious diseases Hepatitis - Infective hepatitis. Sexually transmitted diseases (syphilis, gonorrhea, HIV). Pneumonia, typhoid, urinary tract infections. Tuberculosis. Leprosy. Malaria. Dysentery (Bacterial and amoebic).
15.	25-06-2020	Ms. Shipra Singhal	Sp3, sp2, sp hybridization
16.	26-06-2020	Dr. Abhishek Kumar	Pharmacology in relation with Pathophysiology and HAP covered till now
17.	29-06-2020	Ms. Lakshmi	History and development of the profession of Pharmacy and Pharmaceutical industry in India.
18.	30-06-2020	Mr. Praveen Dixit	Apoptosis and Necrosis.

19.	01-07-2020	Mr. Sanjeev Chauhan	Handling of prescriptions, labeling instructions for dispensed products
20.	02-07-2020	Ms. Garima Kapoor	Hexose monophosphate shunt [HMP shunt]. Diseases associated with carbohydrate metabolism.
21.	03-07-2020	Dr. Roma Ghai	Neoplastic diseases
22.	04-07-2020 (Test)	Dr. Abhishek Kumar	Test from syllabus covered in last 15 days
23.	06-07-2020	Ms. Shipra Singhal	Sigma & pi- bonds, bond lengths, bond angles & bond energies along with their significance in reactions should be carried out.
24.	07-07-2020	Dr. Abhishek Kumar	Pharmacology in relation with Pathophysiology and HAP covered till now
25.	08-07-2020	Ms. Lakshmi	A brief review of present Indian Pharma. Industry in global perspective.
26.	09-07-2020	Mr. Praveen Dixit	Pathogenesis of inflammation. Chemical mediators in inflammation
27.	10-07-2020	Mr. Sanjeev Chauhan	Enlarging and reducing formula, displacement value.
28.	13-07-2020	Ms. Garima Kapoor	Definition. Classification, structures [except B12] biochemical role, sources, daily requirements, & deficiency symptoms.
29.	14-07-2020	Dr. Roma Ghai	Laboratory tests for Liver function tests and kidney function tests
30.	15-07-2020	Ms. Shipra Singhal	Bond polarization, hydrogen bonds, inductive effects
31.	16-07-2020	Dr. Abhishek Kumar	Basics of Pharmacology
32.	17-07-2020	Ms. Lakshmi	Introduction to Pharmacopoeia with reference to IP, BP, USP and International Pharmacopeia.
33.	18-07-2020 (Test)	Dr. Abhishek Kumar	Test from syllabus covered in last 15 days
34.	20-07-2020	Mr. Praveen Dixit	Pathogenesis of chronic inflammation. Repair of wounds in the skin, factors influencing healing of wounds.
35.	21-07-2020	Mr. Sanjeev Chauhan	Preparations of formulations involving allegation, alcohol dilution.
36.	22-07-2020	Ms. Garima Kapoor	Vitamins as co-factors in biochemical reactions.
37.	23-07-2020	Dr. Roma Ghai	Disorders of fluid, electrolyte and acid-base balance
38.	24-07-2020	Ms. Shipra Singhal	Resonance, and hyperconjugation
39.	27-07-2020	Dr. Abhishek Kumar	Pharmacology in relation with Pathophysiology and HAP covered till now
40.	28-07-2020	Ms. Lakshmi	Classification of information, primary, secondary and tertiary. Nomenclature of the drug.

41.	29-07-2020	Mr. Praveen Dixit	Immune response: Specific immunity & immune response Humoral immunity antibody response, mediators of Humoral immunity, basic structure of antibody, antibody classes & functions, maturation of immune response, immunologic memory. Antigens: specificity & Immunogenicity, Natural vs. Artificial Antigens, Soluble, cellular antigens, thymus independent antigen, adjuvant.
42.	30-07-2020	Mr. Sanjeev Chauhan	Introduction to dosage forms. New drug and dosage form. The desirable properties of a dosage form, the need of dosage form.
43.	31-07-2020	Ms. Garima Kapoor	Different types of proteins. Their functions, digestion & absorption. Denaturation & its effect on biological activity. Renaturation of proteins.

Annexure-III

Classes Schedule (Class: 3rd Year)

S. No	Date of the Lecture/ Test	Assigned Faculty	Topics
1.	08-06-2020	Mr. Surya Prakash	Orientation of GPAT Class
2.	09-06-2020	Dr. Abhay Bhardwaj	Introduction to medicinal chemistry
3.	10-06-2020	Dr. Deepti Katiyar	Introduction to Pharmacognosy
4.	11-06-2020	Dr. Mandeep Kumar Arora	Introduction of hemodynamics and Electrophysiology of heart. Anti-hypertensive drugs, Anti-anginal agents.
5.	12-06-2020	Ms. Vidhu Saxena	General Pharmacology || Pharmacokinetics Topic to be covered: 1. Concept of Pharmacokinetics 2. Routes of Administration 3. Definition of Absorption 4. Mechanism of Absorption
6.	15-06-2020	Mr. Anuj Pathak	Introduction to Pharmaceutics
7.	16-06-2020	Mr. Debaprasad Ghosh	1. Suspensions 2. Emulsions
8.	17-06-2020	Ms. Monika Kaurav	Microbiology|| Introduction to Microbiology Topic to be covered: (1)Scope and application to pharmacy field. (2) Whittaker's Five Kingdom concept (3) historical development - biogenesis Vs. abiogenesis,
9.	18-06-2020	Mr. Surya Prakash	Classification of Organic Compounds
10.	19-06-2020	Dr. Abhay Bhardwaj	Benzodiazepine
11.	20-06-2020 (Test)	Mr. Surya Prakash	Test Conducted
12.	22-06-2020	Dr. Deepti Katiyar	Introduction to Phytoconstituents
13.	23-06-2020	Dr. Mandeep Kumar Arora	Anti-arrhythmic drugs, Drugs used in congestive heart failure.
14.	24-06-2020	Ms. Vidhu Saxena	General Pharmacology || Pharmacokinetics Topic to be covered: 1. Factors affecting Absorption 2. Bioavailability
15.	25-06-2020	Mr. Anuj Pathak	Tablets

16.	26-06-2020	Mr. Debaprasad Ghosh	1. Suppositories 2. Semisolids
17.	29-06-2020	Ms. Monika Kaurav	Microbiology|| Introduction to Microbiology Topic to be covered: (1)Germ theory of fermentation (2)Germ theory of disease (3) the contribution of Leeuwenhoek, Robert Koch, Jenner, Louis Pasteur and Ehrlich.
18.	30-06-2020	Mr. Surya Prakash	Nomenclature of Organic Compounds
19.	01-07-2020	Dr. Abhay Bhardwaj	Sedative and Hypnotics
20.	02-07-2020	Dr. Deepti Katiyar	Alkaloid & Glycoside
21.	03-07-2020	Dr. Mandeep Kumar Arora	Anti-hyperlipidemic drugs. Drugs used in the therapy of shock.
22.	04-07-2020 (Test)	Mr. Surya Prakash	
23.	06-07-2020	Ms. Vidhu Saxena	General Pharmacology || Pharmacokinetics Topic to be covered: 1. Drug Distribution 2. Factors affecting drug distribution 3. Volume of Distribution
24.	07-07-2020	Mr. Anuj Pathak	Methods of Formulations
25.	08-07-2020	Mr. Debaprasad Ghosh	1. The Pharmacy Act 1948 (inclusive of recent amendments). 2. Drugs and Cosmetics Act 1940, Rules 1945, including New Drug applications.
26.	09-07-2020	Ms. Monika Kaurav	The contribution of Leeuwenhoek, Robert Koch, Jenner, Louis Pasteur and Ehrlich.
27.	10-07-2020	Mr. Surya Prakash	Stereochemistry of Organic chemistry
28.	13-07-2020	Dr. Abhay Bhardwaj	Sedative
29.	14-07-2020	Dr. Deepti Katiyar	Volatile oil, carbohydrate, fat
30.	15-07-2020	Dr. Mandeep Kumar Arora	Haematinics, anticoagulants and haemostatic agents.
31.	16-07-2020	Ms. Vidhu Saxena	General Pharmacology || Pharmacokinetics Topic to be covered: 1. Binding of drugs to tissue components 2. Metabolism

			3. Types of metabolic reactions
32.	17-07-2020	Mr. Anuj Pathak	Equipment's used in tablet manufacturing.
33.	18-07-2020 (Test)	Mr. Surya Prakash	
34.	20-07-2020	Mr. Debaprasad Ghosh	Pharmaceutical Aerosols
35.	21-07-2020	Ms. Monika Kaurav	Microbiology|| Microscopy and staining technique: Topic to be covered: (1)Electron microscopy (2) The concept of resolving power, Magnification power, numerical aperture and angular aperture and working distance. The principle application of oil immersion microscopy.
36.	22-07-2020	Mr. Surya Prakash	Alkane & Alkene
37.	23-07-2020	Dr. Abhay Bhardwaj	Local Anesthetics
38.	24-07-2020	Dr. Deepti Katiyar	Resin & Tannin
39.	27-07-2020	Dr. Mandeep Kumar Arora	Fibrinolytics and antiplatelet drugs. Blood and plasma volume expanders
40.	28-07-2020	Ms. Vidhu Saxena	General Pharmacology || Pharmacokinetics Topic to be covered: 1. Conversion by Metabolism (active to inactive, active to active, inactive to active/Prodrug) 2. Definition of Excretion 3. Renal and Non-Renal Excretion 4. Rate of Excretion 5. Renal Clearance
41.	29-07-2020	Mr. Anuj Pathak	1. An Introduction to Standard Institutions and Regulatory Authorities such as BIS, ASTM, ISO, TGA, USFDA, MHRA, ICH, WHO. 2. Indian Pharmaceutical Industry- An Overview.
42.	30-07-2020	Mr. Debaprasad Ghosh	1. Pharmaceutical Aerosols 2. Liquids
43.	31-07-2020	Ms. Monika Kaurav	Microbiology|| Microscopy and staining technique: Topic to be covered: (1) Theory of staining, principle and technique of staining procedure - Monochrome, (2) Gram, acid-fast, (3) negative, capsule, endospore.

Annexure-III

Classes Schedule (Class: 4th Year)

S. No	Date of the Lecture/ Test	Assigned Faculty	Topics
1.	08-06-2020	Dr. Puspendra Kumar	How to prepare for GPAT and Orientation Class
2.	09-06-2020	Dr. Ashok Jangra	General anesthetics. Alcohols and disulfiram. Sedatives, and hypnotics
3.	10-06-2020	Dr. Ashu Mittal	The fate of drug after drug absorption, various mechanisms for drug absorption
4.	11-06-2020	Mr. Himanshu Aggarwal	Introduction to microbiology Classification of micro-organisms (Bacteria and Virus) Staining techniques Applications of micro-organisms (Sources of anti-biotics)
5.	12-06-2020	Dr. K. Nagarajan	Carbohydrates and their metabolic pathways
6.	15-06-2020	Mr. Shadab A. Siddiqui	General principles: A brief review of classification & sources of organic compounds, sp3, sp2, sp hybridization, sigma & pi- bonds, bond lengths, bond angles & bond energies along with their significance in reactions should be carried out.
7.	16-06-2020	Dr. Parul Grover	Sedative hypnotic (Medicinal Chemistry)
8.	17-06-2020	Ms. Monika Bhardwaj	Anticancer Drugs
9.	18-06-2020	Ms. Richa Goel	Volatile Oils
10.	19-06-2020	Ms. Kiran Sharma	Micromeritics
11.	20-06-2020 (Test)	Dr. Puspendra Kumar	Mock Test
12.	22-06-2020	Dr. Puspendra Kumar	Chromatography-I
13.	23-06-2020	Dr. Ashok Jangra	Centrally acting muscle relaxants, Psychopharmacological agents: Antipsychotics, antidepressants
14.	24-06-2020	Dr. Ashu Mittal	Compartment Pharmacokinetics
15.	25-06-2020	Mr. Himanshu Aggarwal	Sterilization (Types and applications) Disinfection and antiseptics (Applications and methods of evaluation)
16.	26-06-2020	Dr. K. Nagarajan	Proteins and their metabolic pathways
17.	29-06-2020	Dr. Parul Grover	General anaesthetics (Medicinal Chemistry)
18.	30-06-2020	Mr. Shadab A. Siddiqui	An overview of bond polarization, hydrogen bonds, inductive effects, resonance, and hyperconjugation be taken. Concept of homolytic & heterolytic bond fission, acidity & basicity with different theories should be covered briefly.

19.	01-07-2020	Ms. Monika Bhardwaj	Antibiotics
20.	02-07-2020	Ms. Richa Goel	Volatile Oils
21.	03-07-2020	Ms. Kiran Sharma	Interfacial phenomena
22.	04-07-2020 (Test)	Dr. Puspendra Kumar	Mock Test
23.	06-07-2020	Dr. Puspendra Kumar	Chromatography-II
24.	07-07-2020	Dr. Ashok Jangra	Antianxiety agents, anti-manics and hallucinogens. Anti-epileptic drugs
25.	08-07-2020	Dr. Ashu Mittal	Post Marketing Surveillance
26.	09-07-2020	Mr. Himanshu Aggarwal	Microbiological assays (Types) Test organism for antibiotic assays Sterility testing
27.	10-07-2020	Dr. K. Nagarajan	Lipids and their metabolism
28.	13-07-2020	Dr. Parul Grover	Cholinergic agents (Medicinal Chemistry)
29.	14-07-2020	Mr. Shadab A. Siddiqui	Ease of formation & order of stabilities of electron deficient & electron rich species along with the reasons for the same should be covered.
30.	15-07-2020	Ms. Monika Bhardwaj	Cardiovascular Drugs
31.	16-07-2020	Ms. Richa Goel	Tannins
32.	17-07-2020	Ms. Kiran Sharma	Rheology
33.	18-07-2020 (Test)	Dr. Puspendra Kumar	Mock Test
34.	20-07-2020	Dr. Puspendra Kumar	Analytical Pharmacognosy
35.	21-07-2020	Dr. Ashok Jangra	Anti-parkinsonism drugs. Nootropics. Narcotic analgesics, drug addiction, drug abuse, tolerance and dependence
36.	22-07-2020	Dr. Ashu Mittal	Development, scale up & post approval changes [SUPAC] &*in vitro* [dissolution] *in vivo* [plasma concentration profile] correlation or IV/IV correlation (IVIVC).
37.	23-07-2020	Mr. Himanshu Aggarwal	Biochemical tests
38.	24-07-2020	Dr. K. Nagarajan	Enzymes & Types of Inhibition including kinetics
39.	27-07-2020	Dr. Parul Grover	Dental Products (Pharmaceutical Inorganic Chemistry)
40.	28-07-2020	Mr. Shadab A. Siddiqui	Relationships between energy content, stability, reactivity & their importance in chemical reactions should be covered. Calculations for determining empirical & molecular formula should be covered.
41.	29-07-2020	Ms. Monika Bhardwaj	Antipsychotics
42.	30-07-2020	Ms. Richa Goel	Tannins
43.	31-07-2020	Ms. Kiran Sharma	Complexation

Annexure-IV

Class Report

S. No.	Details	Remarks
1.	Topic	
2.	Detailed Subtopics	
3.	Date of the Class	
4.	Class Duration	
5.	Number of Students Present	
6.	Name of the Absent Students	
7.	Major Outcome	
8.	Any Other Details	

Signature of the Faculty: ---

Signature of the Co-ordinator: ---

Appendix-A

11th Oct'21

Office Order: 01/2021 - Restructured Team -Office of Dean Academics
(Revision in few Appointments)

AMENDMENT NOTE

Ref: Office Order 01/2021 titled Restructured Team - Office of Dean Academics Revision in few Appointments @ Para 4.4.2 of Annexure-X to Appendix-A w.r.t. NIRF & Student's Club (GPAT) Coordinator.

1. This is to notify that **following revision** in few appointments made in the Office Order under reference (Copy attached).

   (a) **Dr. Prakash Srivastava, Associate Professor (CSE) & NIRF Coordinator (Engineering)** is hereby appointed as **NIRF Coordinator (Institute Level).**

   (b) **Dr. Vipin Kumar, Professor (AS)** is designated as **NIRF Coordinator (Research).**

   (c) **Mr. Surya Prakash, Asst. Professor (KSOP)** is appointed as **GPAT Coordinator** (Student's Club).

2. The **appointees at Para 1 (a-c)** above shall perform their duties in designated capacity in addition to their existing academic/administrative responsibilities.

3. The above comes into force with immediate effect.

4. There is no other change in Office Order under reference.

Dr. (Col) A Garg
Director
11.... Oct'21

Distribution:

- Joint Director/ Deans/ Principal-KSOP/ HoDs (CSE/IT/CS/CSIT/EC/EN/CE/ME/AS/MCA/MBA/HS), COE, Dy. GM-TBI, Head-CRPC, Head-CAM/CAW, Head-AEC, Head-IRCDC, Head-EC (IEC), Addl. Head-IIPC, Chairperson-ICC, Head IT Operations, Head PR & IR, Head-HR, Accounts Officer, Registrar, Admin Officer, Librarian, Purchase Officer, KIETians

Copy to:

- Dean A & Team
- Dr. Prakash Srivastava (CSE), **Coordinator- NIRF (Engineering & Institute Level)**
- Dr. Vipin Kumar (AS), **Coordinator - NIRF (Research)**
- Mr. Surya Prakash (KSOP), **Coordinator - GPAT (Student's Club)**

Ref/ Dir/ 17/ 2020 **Date: 11th Sep'20**

OFFICE ORDER: 17/2020

Formation of Students Higher Education Preparation Club

1. **Students Higher Education Preparation Club** is hereby constituted with an objective to support students in their preparation towards various national level exam e.g GATE, CAT etc. To start with the club will prepare students for **GATE (Graduate Aptitude Test in Engineering) -** a gateway for admission to the various postgraduate engineering programs offered by the IITs, NITs, IIITs as well as Public Sector Undertaking (PSUs) jobs. **In a phased manner the club will commence preparation for other examinations as well depending on the need of students.**

2. A well-structured policy doc. covering the objectives, Structure of Club involving Chief Coordinator, Departmental Coordinators, Faculty mentors & Student Coordinators core committee, their roles & responsibilities and separate guidelines for student members of the club, Questions & marking scheme of GATE, Preparation strategy, Study support - resources, Targets etc. is attached for a detailed understanding and clarity **(Appendix-A).** Similar guidelines for preparation of other examinations like CAT etc. shall be issued separately on need basis & **Appendices duly marked shall be added**.

3. **Dr. Arunesh Chandra, Professor (ME)** is hereby designated as **Chief Coordinator - Students Higher Education Preparation Club** supported by **Departmental Coordinators, Faculty mentors & Student Coordinators core committee** under the overall direction and support of **Dean (A).** He shall be performing the above duties in addition to his regular academic engagement.

4. The overall responsibility of running the **club's activities smoothly @ department level** shall be with **departmental Coordinators & respective HoDs.** The entire team associated with **Students Higher Education Preparation Club** at **para 3** shall be responsible to **Chief Coordinator** for their duties in their designated capacities.

5. **Dr. Arunesh Chandra** shall submit monthly progress *w.r.t.* Club's periodic progress/ time table / planning etc. for GATE & other examination's preparation to **Dean (A)** on regular basis without fail.

Dr. (Col) A Garg
Director

Distribution:

- Joint Director/ Deans/ Principal-KSOP/ HoDs (CSE/IT/EC/EN/EI/CE/ME/AS/MCA/MBA/HS/CS/CSIT)/ GM-TBI & Dean IEC, Head-CRPC, Head-CAM, Head-AEC, Head-IRCDC, Head-IC, Addl. Head-IIPC, Chairperson-ICC, Head-IT Operations, Manager-IA, Head-HR, Registrar, Admin Officer, Accts Officer, Librarian, Purchase Officer, KIETians, Students

Copy to: Dr. Arunesh Chandra & all concerned members of Club & Dean A

Appendix-A
(GATE Preparation)

Students Higher Education Preparation Club

1. Introduction:

- GATE (Graduate Aptitude Test in Engineering) exam is a gateway for admission to the various postgraduate engineering programs offered by the IITs, NITs, IIITs as well as Public Sector Undertakings (PSUs) jobs.
- Many other premier institutes apart from the IITs, NITs, IIITs and GFTIs also consider GATE score for offering admissions to the candidates.
- GATE examination is conducted jointly by the Indian Institute of Science (IISc), Bangalore and the seven Indian Institutes of Technology on behalf of the National Coordination Board (NCB)-GATE, Department of Higher Education, MHRD, Government of India.
- The GATE score reflects the relative performance level of the candidate in a particular discipline, which is quantified based on the several years of examination data.
- GATE qualified candidates are also eligible for the award of Junior Research Fellowship (JRF) in CSIR Laboratories and CSIR sponsored project.

2. Objective:

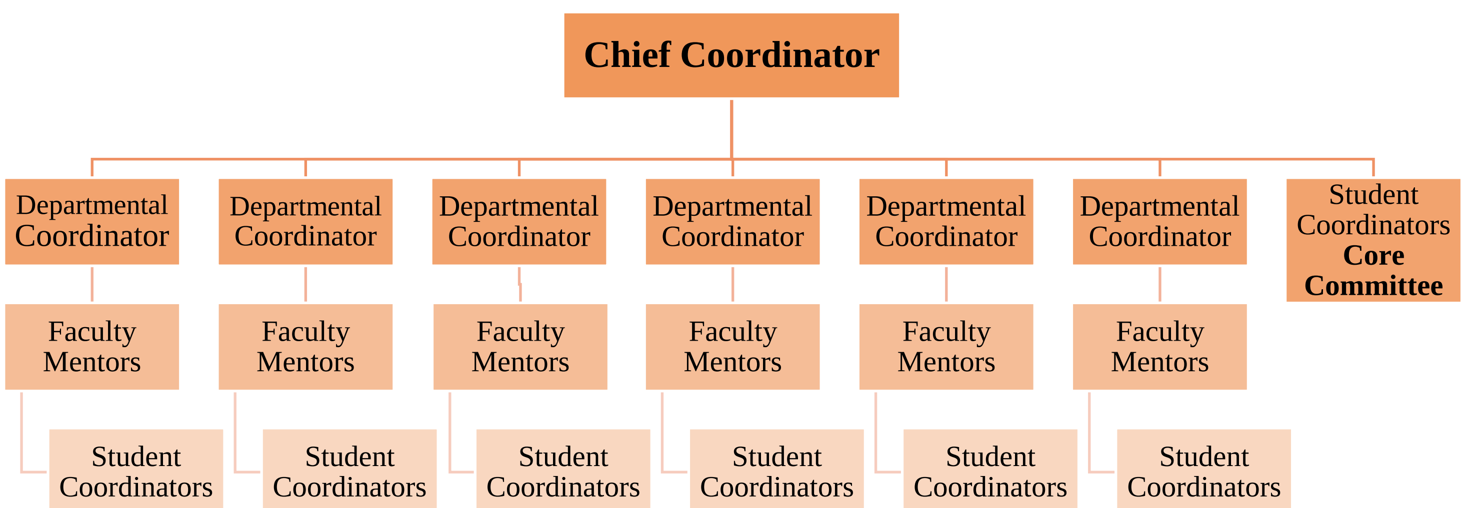
- It is very much evident from the experience that GATE is a conceptual exam and having a great command over concepts is necessary for clearing GATE examination. For cracking an examination like GATE, a lot of preparation with firm determination is required. Students Higher Education Preparation Club constituted at KIET Group of Institutions will provide a platform for GATE aspirants for strengthening of concepts, proper coverage of complete syllabus, discussion, doubt resolution, and maximum practice.

- Since long-time elapses before one actually faces the GATE exam, it is important to keep the preparation going in full flow, the similar way as it started initially. The club will allow students to excel in their domain and to provide rigorous training to registered students under guidance of faculty mentors/subject expertsfor GATE exam daily after the college hours (4:00 to 6:00 pm) from 2nd year onwards of B. Tech. program.

3. Scope:

- **Strengthening of Concepts-** The club will help in formation of groups, consisting of aspirants with similar goals, successful senior/aspiring senior under guidance of faculty mentors, who can help in strengthening of concepts. It is well known that no one can succeed alone; one cannot go long if one plans to go alone. Taking help and guidance from fellow aspirants/senior aspirants and faculty mentor is, thus, always a good idea.
- **Rigorous Practice-** How to interpret a question, how to approach it, how to verify the answer without re-solving it, maintaining accuracy even in a pressure situation, all such skills are needed for GATE. The club will act as a platform where like-minded students with similar goals can share and discuss their problems and doubts. A number of aspirants may connect here to find useful resources, discuss and seek answers even to their technical doubts.
- **Proper Guidance-** The club will be providing guidance and support to help students to resolve their doubts, both technical and preparation related. It will help them in formulating strategy for preparation and personal follow up. The mentors will help the GATE aspirants to stay motivated and committed throughout the preparation.

4. Structure of the Club



S. N.	Name	Appointment	Department
1	Dr. Arunesh Chandra	**Chief Coordinator**	Mechanical Engineering
2	Dr. Sanjeev Singh	**Departmental Coordinators**	Civil Engineering
3	Mr. Ajay Kumar		Mechanical Engineering
4	Dr. Varun Gupta		Electronics & Instrumentation Engineering
5	Mr. Hriday Gupta		Computer Science Engineering / Computer Science
6	Mr. Satya PSingh		Electronics & Communication Engineering
7	Mr. Shiv K Sikarwar		Electrical & Electronics Engineering
8	Mr. Vijay Singh		Information Technology / Computer Science & Information Technology
9		**Faculty Mentors**	Assigned by concerned departments
1	Aakash Kumar (III yr.-A) aakash.1822co1001@kiet.edu	**Student Coordinators Core Team**	Computer Science
2	Subhav Gaur (III yr.-B) subhav.1822ce1082@kiet.edu		Civil Engineering
3	Darshika Agarwal (III yr.-A) darshika.1822csi1010@kiet.edu		Computer Science & Information Technology
4	Arunesh K Pandey (III yr.-A) arunesh.1822me1033@kiet.edu		Mechanical Engineering
5	Yashraj Srivastva (III yr.-C) yashraj.1822en1168@kiet.edu		Electrical & Electronics Engineering
6	Gargi Agarwal (III yr.-A) gargi.1822ec1058@kiet.edu		Electronics & Communication Engineering
		Student Coordinators (Year/Section wise)	**List attached (Annexure-1), P-14**

5. Roles and Responsibilities:

Chief Coordinator

- Responsible to take the decisions for the club.
- Preparation of detailed working and implementation plan for the club.
- Responsible for **reporting of monthly activities to Dean Academics** for perusal by JD/ Director.

- To take feedback from Students and Faculty mentors fortnightly, analyze them and to take necessary initiatives for smooth functioning.
- To monitor and maintain discipline.
- Smooth coordination with departmental Coordinators, Faculty Mentors, Student Coordinators
- Conduction of Mock Test, Declaration of Result and Analysis of Result periodically.
- Preparation of various forms like registration form, feedback form, Class report form etc.

Departmental Coordinator

- Coordination with Chief Coordinator, Faculty Mentors, Student Coordinators for smooth functioning.
- Preparation of Class Schedule.
- Responsible for design and smooth conduction of the club activity at department level.
- Conduction of Mock Test, Declaration of Result and Analysis of Result periodically at department level.
- Responsible for taking feedback of students on regular basis.
- Responsible to discuss the feedback of students with Chief Coordinator.

Faculty Mentors

- Responsible for smooth conduction of classes.
- To resolve s doubts, both technical and preparation related
- Month-wise and subject-wise strategy for preparation
- Daily study plan and personal follow-up
- To keep students motivated and committed throughout the preparation
- Conduction of open discussion among group of students on the covered topics
- Practice tests after completion of topics to improve the understanding of topics and pattern of exam.
- Preparation of MCQs of the respective subject/topic.

Student Coordinators

- Coordination with Departmental Coordinator, Faculty Mentors and Students for smooth functioning of club.
- To coordinate and support in organization of various club related activities.

- Responsible for ensuring that all the affairs of a club are carried out in a smooth and transparent manner.
- The Student Coordinator has to make necessary arrangements for the club meetings. These include deciding the date, time of the meeting and sending invitations to club members. The Student Coordinator will be taking down the minutes of a meeting.
- They will directly communicate with the club members/ students of respective class. Welcomes new members and invite them to club/group activities.

6. Guidelines for Student Members of the Club:

- The interested students of second and third year, who have decided their career prospects, are eligible for the membership of club. Interested students should be motivated to register themselves at department.
- Most of the basic technical subjects asked in the examination derive their concepts from the technical subjects that are taught in the third or fourth semester of one's bachelor's degree. This is why the second year of graduation would be the right time to start GATE preparation.
- Starting at such an early stage also leaves ample scope for strengthening of concepts, proper coverage of complete syllabus, discussion, doubt resolution, and maximum practice.
- The GATE Classes will run daily after the college hours (4:00 to 6:00 pm) from 2nd year onwards of B. Tech. program.
- Based on the past performance of the students of individual departments of the Institute the target of the GATE club of the institute should be fixed. Every department should run this club separately and students enrolled per batch should be limited initially, and if more students are interested then more batches can be started.
- Students giving their consent to attend GATE classes must ensure more than 90% attendance in GATE classes. Student fails to achieve the attendance criteria will be debarred from the GATE classes. For some genuine reasons, the attendance criteria can be lowered to 80%.
- Any student wants to join GATE classes in their 7th semester; he/she must be required to score minimum 60% in three consecutive MOCK tests to be arranged by concerned department.

- If student is not able to score more than 40% in three-consecutive tests, he/she will not be eligible to continue classes.
- All classes shall be free of cost for all the interested students.
- GATE guidelines as applicable shall be given time to time by faculty members teaching that subject and GATE coordinator of Department.
- Students can discuss the problems with the Student Coordinators / faculty Coordinators / faculty mentors.
- Weekly mock test series will be arranged to assess the performance of students, it will be continuously monitored, and target should be to have most students achieve AIR less than 1000.
- Concerned subject faculty will open discussion among students on the covered topics and test questions. This will improve the understanding of topics and pattern of exam. Separate revision classes will be arranged for already taught courses from time to time.
- Students will take part in feedback of the teaching-learning process and help to improve the quality of teaching. Feedback will be taken from the students fortnightly.
- For more conceptual learning, The GATE students must utilize the books and study materials available with the department library/Central library.
- An open discussion forum should be there between student representatives including core team, Chief Coordinator & Departmental Coordinators so that the need of the students can be addressed easily, effectively and speedily.

7. Questions and marking scheme of GATE:

The examination will consist of totally 65 questions, segregated as One-mark and Two-mark questions. Out of 65 questions, 10 questions will be from General Aptitude (Verbal and Numerical ability) and 55 questions will be Technical, based on the paper (branch) chosen. The General Aptitude section will have 5 One-mark questions and 5 Two-mark questions, accounting for about 15% of total marks. The Technical section and Engineering Mathematics section will total have 25 One-mark questions and 30 Two-mark questions, accounting for about 85% of total marks. Further, all the sections will have both Multiple-Choice Questions (MCQs) and Numerical Answer Type questions (NATs). The examination awards negative marks for wrong MCQ answers. Usually, 1/3rd of original marks will be deducted for wrong MCQ answers (i.e. -0.33 for wrong One-mark answers and -0.66 for wrong Two-mark answers) while there are no negative marks for NATs.

Multiple Choice Questions (MCQs)

These questions are objective in nature and each question will have choice of four answers, out of which the candidate has to mark the correct answer. Each question carries 1 or 2 marks questions in all the sections.

Numerical Answer Questions

There will be no choices available for these types of questions. A Numeric Answer question carries 1 or 2 marks questions in all sections. The answer for these questions is a real number to be entered by using mouse and virtual keypad displayed on the monitor. No negative marking for these questions.

Negative Marking Scheme

Negative Marking		
Question Type	**Marks**	**Negative Marking**
Multiple Choice Questions (MCQs)	1 mark	0.33
	2 marks	0.66
Numerical Answer Questions	1 mark	NIL
	2 marks	NIL

8. Strategy for GATE Preparation:

In order to achieve a better score in GATE there should be a perfect timetable for exam preparation. The timetable for GATE exam can be prepared by considering the following factors:

- After checking the GATE syllabus, there should be clarity on the subjects or topics (strong and weak).
- Probably two subjects should be taught in a month for 2 hours per day.
- Aim to cover whole subject in a given time frame
- Faculty assigned to teach particular subject should review list of topics to be discussed.
- Estimation of time required to prepare each topic/subject.
- Depending on that, prepare a monthly plan for GATE preparation.

Normal duration of teaching may be from 4:00 pm to 6:00 pm from Monday to Friday for 2-hour duration and Saturday can be used for mock test. Target for two subjects to be completed is approximately 1 month. Department can frame or tune the timetable as per their feasibility, but it is to be ensured that the last two months of GATE exam preparation should be completely allocated to the revision of syllabus. The following timetable is just for reference purpose, and it is not mandatory to follow the same timetable for exam preparation.

Sample Monthly Time-table

Day	Subjects	Preparation Strategy
Day 1 to 5	Subject 1	Learn all the concepts and solve relevant problems from Chapters 1 to 5.
Day 6	Subject 1	Revise the learnt topics from Chapter – 1 to 5. Practice mock test and analyse performance of students.
Day 7	REST	
Day 8 to 12	Subject 1	Learn all the concepts and solve relevant problems from Chapters 6 to10.
Day 13	Subject 1	Revise the learnt topics from Chapter – 6 to 10. Practice mock test and analyse performance of students.
		Providing database of previous 5 years question papers along with solutions. Feedback of the subject 1
Day 14	REST	
Day 15 to 19	Subject 2	Learn all the concepts and solve relevant problems from Chapters 1 to 5.
Day 20	Subject 2	Revise the learnt topics from Chapters – 1 to 5. Practice mock tests and analyse performance of students.
Day 21	REST	
Day 22 to 26	Subject 2	Learn all the concepts and solve relevant problems from Chapters 6 to 10
Day 27	Subject 2	Revise the learnt topics from Chapter – 6 to 10. Practice mock tests and analyse performance of students.
		Providing database of previous 5 years question papers along with solutions. Feedback of the subject 2
Day 28	REST	
Day 29	Doubt Clearing Session on previous 5 years question papers of subject 1	
Day 30	Doubt Clearing Session on previous 5 years question papers of subject 2	

Importance of GATE Mock Test

- Mock tests will help the applicants revise the entire syllabus and clear their basic concepts.
- By practicing mock tests, students will be able to finish their actual paper in time.
- Another advantage of practicing the GATE mock test is that students will be able to identify their weaknesses and strengths.
- Taking mock tests also reduces the fear of examination.
- Taking mock tests can help improve candidates' understanding of the nature of questions, and they also get an idea of how much time does it take to solve each section.

Importance of Revision

- Having good concepts is a necessary condition but regular practice is a must for cracking GATE.
- Most questions asked in GATE are numerical problems and not theoretical ones. So, if one wants to crack it, merely knowing theoretical concepts is not sufficient. One must know how to apply it.
- Students must give enough time to revise various topics and chapters so that everything stays fresh in the mind.
- This will also allow student to know how much of the topics and chapters that they have studied have retained.
- Revision should be done regularly.

9. Resources

Faculty Mentors

It is assumed that in GATE examination, there may be 15-16 subjects required to be covered in technical section apart from Engineering Mathematics and General Aptitude. Since faculty mentors will be actively involve themselves in guiding students for GATE and other technical competitive examinations, the involvement should be considered and counted as normal teaching load. Depending upon the expertise and experience, faculty mentors should be nominated to teach particular subject(s). The nominated faculty mentors should be eligible for flexible timings.

Student Mentors

Student Mentors will be the moderator for the student discussion forum, so that the openness of students may be increased, and can take some classes depending upon their interest and availability. These mentors can moderate open question answers session, it will increase the interactions and concept understanding among the students during and after the classes. Student Mentors can help with doubt resolution of fellow club members.

Study Materials

Institute should purchase sufficient number of quality contents for delivering lectures (past few year papers or some coaching contents) for better practice of students. Since standard reference books of GATE contains numerous concepts and topics, and some of them are very relevant as per prescribed GATE syllabus. Study material by MADE EASY is very thorough, precise and written in a language easy for the students to understand. Any Good test series Made Easy, ACE academy, IES masters are preferred and can be purchased by the Institute as reference material for students.

Sample Flow Chart of Time Plan of Gate Club for AY 2020-21

Registration of students of second/third year in GATE Coaching Club by 15th August 2020

↓

Commencement of classes from 17th August 2020

↓

Target from 17th August 2020 to 5th December 2020 (to study subjects of II, III and IV semesters) relevant to GATE for particular branch

↓

Break of one month as per University semester examination (probably from 2nd week of December 2020 to 1st week of January 2021)

↓

Target from 2nd week of January 2021 to 8th May 2021 (to study remaining any subjects of previous and current semesters)

↓

Break of one month as per University semester examination (probably from 2nd week of May 2021 to 1st week of June 2021)

↓

Commencement of classes from 7th June 2021 to 30th November 2021 for GATE preparation of remaining subjects

↓

Revision classes/MOCK tests from December 2021 to probably February 2022 before final date of GATE examinations as announced in GATE form

Registration Form

Name of the Student	
University Roll No.	
Father's Name	
Batch	
Section	
KIET Mail ID	
Mobile Number	
Commitment towards GATE Classes	
Declaration	**I am interested in GATE Preparation Classes. I will maintain my attendance in the GATE Classes as per norms. I have read the terms and conditions for GATE Classes and I am giving my consent to abide by these conditions**
Signature of the participant	

Class Schedule

Class		2^nd^ Year/ 3^rd^ Year/ 4^th^ Year	
S. No.	**Details**	**Planned**	**Conducted**
1.	**Total Number of Classes**		
2.	**Total Number of Tests**		
3.	**Total Number of Open Discussions**		
4.	**Student's Feedback**		
5.	**Faculty remarks**		
6.	**Coordinator Remarks**		
7.	**HoD Remarks**		
8.	**Signature of HoD**		

Class Report

S. No.	Details	Remarks
1.	**Topic**	
2.	**Detailed Subtopics**	
3.	**Date of the Class**	
4.	**Class Duration**	
5.	**Number of Students Present**	
6.	**Name of the Absent Students**	
7.	**Major Outcome**	
8.	**Any Other Details**	

Signature of the Faculty: --

Signature of the Coordinator: --

Annexure-1

Nomination of Student representative for GATE club

S.NO	BRANCH	YEAR	Section	NAME OF STUDENT	Email	Contact
1	CSE	2	B	NISHANT PRATAP SINGH	nishant.1923cs1141@kiet.edu	8126973714
2	CSE	2	A	ADARSH GUPTA	Adarsh.1923cs1036@kiet.edu	6393112637
3	CS	2	A	KESHAV BHARDWAJ	keshav.1923co1114@kiet.edu	9823992888
4	CS	2	A	GARVIT PUNDIR	garvit.1923co1054@kiet.edu	9667779783
5	CSE	3	B	HITESH AGARAWAL	hitesh.1822cs1187@kiet.edu	7800417686
6	CSE	3	A	AARUSH GANDHI	aarush.1822cs1177@kiet.edu	8077807979
7	CS	3	A	KANIKA KANSAL	kanika.1822co1036@kiet.edu	9667987551
8	CE	2	A	Kratika Mishra	kratika.1923ce1017@kiet.edu	7007046822
9	CE	2	B	Shashwat Singh	shashwat.1923ce1070@kiet.edu	7037097001
10	CE	3	A	AMAN SENGAR	aman.1822ce1009@kiet.edu	9568911286
11	IT	2	A	Durgesh Agrhari	durgesh.1923it1156@kiet.edu	9554510307
12	IT	2	B	Komal Sharma	komal.1923it1199@kiet.edu	7827723405
13	IT	3	A	Ayush Parashar	ayush.1822it1041@kiet.edu	9264971540
14	IT	3	C	Raghav Goel	raghav.1822it1180@kiet.edu	8923845703
15	CSIT	2	B	Ritik Bhardwaj	ritik.1923csi1009@kiet.edu	9084962436
16	CSIT	2	A	Harsh Dayal	harsh.1923csi1065@kiet.edu	9643285239
17	CSIT	3	A	Prafull Varshey	Prafull.1822csi1030@kiet.edu	8533983256
18	ME	3	D	Somit Shivhare	somit.1822me1137@kiet.edu	9621105479
19	ME	2	B	Manjul Mishra	manjul.1923me1044@kiet.edu	9695160909
20	ME	2	C	Vishal Singh	vishal.1923me1106@kiet.edu	9026414201
21	EN	2	A	Alok kumar	alok.1923en1131@kiet.edu	7309483201
22	EN	2	C	Sonal Mishra	sonal.1923en1082@kiet.edu	6390195836
23	EN	3	A	Bhavishya Tomar	bhavishya.1822en1043@kiet.edu	6398183316
24	EC	2	A	Bhavya Aggarwal	bhavya.1923ec1066@kiet.edu	7078305307
25	EC	2	C	Tanya Agarwal	tanya.1923ec1120@kiet.edu	6396742796
26	EC	3	A	Apoorva Chand	apoorva.1822ec1038@kiet.edu	6394557884
27	EI	3		Nikita Ray	nikita.1822ei1017@kiet.edu	8265805563
28	EI	3		Vishwas Garg	vishwas.1822ei1030@kiet.edu	8743989928

Ref/ Dir/27/2021

9th Oct'21

OFFICE ORDER: 27/2021

Skill Enhancement Programme for Smart Empowered Professionals (SEP)
(Revised - Policy on Training)

Reference: Office Order 11/2019 dated 12 Jun'19

1. It is hereby informed that policy under reference has been revised to accommodate some changes **(Refer - Annexure 1 for revised policy)** to be implemented w.e.f. forthcoming semester i.e., **academic year 2021-22** spanning the entire stay of the student in the campus. The programme will be as per the following duration: B. Tech (200+ Hrs.), B. Pharm (60+ Hrs.), MCA (100+ Hrs.), MBA (90+ Hrs.) and duly integrated in timetable. Semester wise distribution is tabulated below:

Name of Program	Sem I	Sem II	Sem III	Sem IV	Sem V	Sem VI	Sem VII
B.Tech	20 hours + 20 for half of the students	20 hours + 20 for the other half of the students	20 hours on QA + LR	20 hours on QA + LR	50 hours	50 hours	Capsule Training as per requirement
B.Pharm	-	-	-	-	20 hours	20 hours	20 hours
MBA	30 hours	30 hours	30 hours	-	-	-	-
MCA	50 hours	50 hours	-	-	-	-	-

2. It can be observed that the training process (Skill Enhancement Programme for Smart Empowered Professionals - SEP) ranges from **60 – 200 + hours** depending on the course (B. Tech / B. Pharm / MCA / MBA) and will consist of the following:

   (a) Training delivery that consists of **'Two trainers in a classroom model for Soft Skills'** for all the streams in the third year of the four-year courses running on campus and for a proportionate time for the other courses. It will also be attempted to freeze at least one trainer per year per section for effective monitoring of the students on a long-term basis. In addition, the trainer's performance may also be ascertained.

   (b) **Assessment:**

   - Internal pre-assessment, mid-assessment, & post-assessment test (of all the training components) will be held for the students of B. Tech 2nd year & pre final year to ascertain their Employability Quotient and monitor their progress. Those students who are found wanting will undergo 20 hrs. of special training under the Winter/Summer School program.
   - Internally, continuous assessment will be practiced through regular classroom activities to account for the outcomes of the Soft Skills training provided; this is also linked to the number of students recruited every year as well as the percentage of training sessions attended by the student.

Director.

3. The **detailed guidelines** along with **topics to be covered in each semester** as per the above policy are attached herewith as appendices:

    (a) Annexure -1 (Training Policy)
    (b) Appendix A - B. Tech 1st year (both semesters)
    (c) Appendix B - B. Tech 3rd semester
    (d) Appendix C - B. Tech 4th semester
    (e) Appendix D - B. Tech 5th semester & MCA 1st semester
    (f) Appendix E - B. Tech 6th semester & MCA 2nd semester
    (g) Appendix F - B. Pharma 5th, 6th & 7th semester
    (h) Appendix G - MBA (1st, 2nd & 3rd semester)

4. The **modalities for implementation** will be as follows:

    (a) The respective HoDs to earmark training classes in the regular timetable - number of hours as per above table in consultation with HOD HS.
    (b) One Soft Skills faculty (Departmental Training Coordinator) will be attached to every department for the smooth coordination of all training related activities for that department.
    (c) The parent department will extend the necessary support to the designated Departmental Training Coordinator for the smooth conduct of all training activities.
    (d) Attendance of students in the sessions to be considered as a part of the regular attendance, for the students to appear for the internal CT's.
    (e) Assessment of training year wise will be done through pre assessment & post assessment test.
    (f) After completion of their pre final year training the students will be handed over to the Finishing School@ KIET for their ASSET Certification.
    (g) If any special measure is required to assist the weak students identified after their preassessment, including the modification of timetable it will be agreed upon mutually, for the benefit of these students.

5. The idea is to fine tune the students' attitude, values, beliefs, futuristic thinking, life-long learning, teamwork, employability, various skill sets of communication, manners, etiquettes etc., to ensure that, they are capable enough to deal with different situations diligently and responsibly and can express themselves clearly, correctly and concisely to contribute towards the betterment of the society, the nation and the world.

6. Please ensure that the policy is implemented in letter and spirit so that the students become **Smart Empowered Professionals** once they graduate from KIET Group of Institutions.

7. With this, the **Office Order 11/2019 dated 12th Jun'19** stands superseded.

Dr. (Col) A Garg
Director
09...Oct'21

Distribution:

- Joint Director/ Deans/ Principal-KSOP/ HoDs (CSE/IT/CS/CSIT/EC/EN/CE/ME/AS/MCA/MBA/HS),COE, Dy. GM-TBI, Head-CRPC, Head-CAM/CAW, Head-AEC, Head-IRCDC, Head-EC/IEC, Addl. Head-IIPC, Chairperson-ICC, Head IT Operations, Head PR & IR, Head-HR, Accounts Officer, Registrar, Admin Officer, Librarian, Purchase Officer, KIETians

ANNEXURE-1

Session-2021-22: Training Policy Implementation Hours & segregation - For all streams & all years: KIET Group of Institutions - Delhi-NCR

Training Verticals / Stream & Semester	Soft Skills	Verbal Ability	Quantitative Aptitude & Logical Reasoning	Communication Improvement (L1 & L2)	Total No. of hours	Comments/Remarks
B. Tech 1st semester	20 hours as per AKTU Audit course + 20 hours as lab sessions for half the sections	NA	NA	NA	60 hours for all – 20 regular and 20 hours extra as lab sessions to all students by taking half of the sections in both the semesters	CT-1 & CT-2 to be done as MCQs & only with subjective type on AKTU pattern
B. Tech 2nd semester	20 hours as per AKTU Audit course + 20 hours as lab sessions for the other half of sections	NA	NA	NA		
B. Tech 3rd semester	NA	NA	20 hours for all	NA	20 for all + 20 hours on Aptitude / Communication on a registration basis	Pre-assessment to be done at the start of 3rd semester; progress check along with CT-2, & mid-assessment at the end along with PUE
B. Tech 4th semester	NA	NA	20 hours for all	NA	20 for all + 20 hours on Aptitude / Communication on a registration basis	Progress check along with CT-2; Post assessment to be done at the end of 4th semester PUE
B. Tech 5th semester	20 hours with two trainers concept	10 hours for all the students	20 hours merged in the regular timetable	NA	50 hours for all	Pre-assessment of all components at the start of 5th sem; progress check along with CT-2; Mid-assessment at the end of 5th sem PUE
B. Tech 6th semester	20 hours with two trainers concept	10 hours for all the students	20 hours merged in the regular timetable	NA	[illegible] hours for all	Progress check along with CT-2; Post Assessment at the end of 6th sem of all the components
B. Tech 7th semester	Capsule training programs / Workshops would be conducted as per the requirement of the company & duration available				Variable	Total Training Hours - B. Tech = 180 hours for all
B. Tech 8th semester	Capsule training programs / Workshops would be conducted as per the requirement of the company & duration available				Variable	
B. Pharm 1st semester	NA	NA	NA	NA	0	
B. Pharm 2nd semester	NA	NA	NA	NA	0	
B. Pharm 3rd semester	NA	NA	NA	NA	0	An Aptitude Winter School (AWS) would be conducted for all the students to enhance their employability - duration 20 hours during the Winter Break
B. Pharm 4th semester	NA	NA	NA	NA	0	
B. Pharm 5th semester	20 hours with two trainers concept	NA	NA	NA	20	
B. Pharm 6th semester	20 hours with two trainers concept	NA	NA	NA	20	
B. Pharm 7th semester	20 hours with two trainers concept				20	Total Training Hours - B. Pharm = [illegible] for all students
B. Pharm 8th semester	Capsule training programs / Workshops would be conducted as per the requirement of the company & duration available				Variable	
MCA 1st semester	20 hours with two trainers concept	10 hours for all	20 hours for all	NA	50 hours for all	Pre-assessment of all components at the start of 1st sem; Mid-assessment at the end of 1st sem
MCA 2nd semester	20 hours with two trainers concept	10 hours for all	20 hours for all	NA	50 hours for all	Post Assessment at the end of 2nd sem of all the components
MCA 3rd semester	Capsule training programs / Workshops would be conducted as per the requirement of the company & duration available				Variable	Total Training Hours - MCA = 100 for all students
MCA 4th semester	Capsule training programs / Workshops would be conducted as per the requirement of the company & duration available				Variable	
MBA 1st semester	10 hours with two trainers concept	10 hours	10 hours	NA	30	An Aptitude Winter School (AWS) would be conducted for all the students to enhance their employability - duration 20 hours during the Winter Break
MBA 2nd semester	10 hours with two trainers concept	10 hours	10 hours	NA	30	
MBA 3rd semester	10 hours with two trainers concept	10 hours	10 hours	NA	30	
MBA 4th semester	Capsule training programs / Workshops would be conducted as per the requirement of the company & duration available				Variable	Total Training Hours - MBA = 90 for all students

NOTE: Winter School/Summer School on all the components of training would be organized on a registration basis as per the availability of students.

(Director)
09 OCT 21

Annexure-1

KIET GROUP OF INSTITUTIONS

(A Technical Campus approved by AICTE &
Affiliated to Dr. A.P.J. Abdul Kalam Technical Univ., Lucknow)
Accredited by NAAC with Grade 'A' (5 yrs)
Ghaziabad-201206

POLICY ON TRAINING – for STUDENTS

Effective communication, interpersonal skills, verbal ability, and aptitude are crucial to compete in a healthy manner in the professional environment. The real key to the effectiveness of professionals is their ability to put their domain knowledge into effective practice. In this context, soft skills, verbal ability, and aptitude have a crucial role to play. If future Managers/Engineers/Pharmacists know how to deal with people at the interpersonal & emotional level by aptly using **Emotional Intelligence (EI) and problem-solving skills to focus on the pursuit of excellence in higher education, for gaining meaningful recruitments characterized by unlimited possibilities with competence to create a global impact.**

Through our soft skills, verbal ability, and core aptitude training, students can unveil the hidden potential lying dormant in them.

Vision of the H & S Department

To enable the seamless integration of our students into the corporate world by shaping their careers through the learning by doing concept, which will ensure their contribution towards Societal, National, and Global development.

Mission of the H & S Department

To empower our students with sound English Communication Skills, and professional acumen by enhancing their competencies through apt mentoring & instilling in them the values, skills & positive attitude which will help them in climbing up the employability ladder.

The development of soft skills is extremely essential in today's competitive corporate arena therefore the Institute conducts exclusive sessions that aim at the enhancement of the employability quotient of the students.

Objectives

To enable our students to face the world with confidence and self-belief while acknowledging the significance of **soft skills, verbal ability, and core aptitude training** for their personal and professional facade.

1. To self-explore & become self-aware thereby leading to self-improvement
2. To hone the communication skills (LSRW) of the trainees
3. To provide a platform to unleash their hidden potential (through E-Boosters, peer learning)
4. To provide insight of the recruitment process & tests (through a glimpse of QA+VA+LR)
5. To develop and strengthen various professional skills such as e-mail writing, presentation, leadership qualities, and team building (interpersonal skills)
6. To create a desire to fulfil individual career goals (CV/Resume/E-Portfolio preparation)
7. To enhance the confidence level of the students through GDs, PIs, & other group activities
8. To make them ready to face the real-life challenges (adapting to change)
9. To give a realistic perspective of work and work expectations (diversity & inclusivity)
10. To ensure a smooth transition from campus to corporate by developing industry and career ready professionals

The following is the plan of action for the **SKILL ENHANCEMENT PROGRAM for SMART EMPOWERED PROFESSIONALS (SEP for SEP)** according to the respective years **from 2K21-22 session** onwards:

First year - SHARPEN - Shaping Attitude and Restructuring the Psyche through Effective English Communication

The students would go through the AKTU prescribed audit course on Soft Skills KNC – 101 in the 1st semester & KNC – 201 in the 2nd semester & their performance would be adjudged according to their result in the subject. Considering, the number of weeks available for both the semesters – the number of training hours would vary between 40 – 48 hours in the first year. Half of the sections will also undergo 2 hour/week of lab/practical sessions to imbibe the required hand-on skills in alternation with that half of the students who undergo sessions on the AKTU prescribed English lab sessions in either semester.

Total number of hours – Soft Skills Training = 60 to 66 for all the students.

Second year – Aptitude Basics

TRAINING PLAN for the batch 2020-24 moving to B. Tech 2nd year in the session 2021-22

All the B. Tech 2nd year students would undergo 20 hours of training on Quantitative Aptitude & Logical Reasoning in both the semesters. An **internal pre-assessment test** would be taken at the start of the 3rd semester, a **mid-assessment test** would be taken along with the PUE of 3rd semester, and a **post-assessment** would be taken along with the PUE of 4th semester to ascertain the progress of the students.

Total number of training hours for all the students = 40 to 44

Third Year: CHASE – Constructive & Holistic Approach to Smooth Employment

TRAINING PLAN for the 2019-23 batch (moving to 3rd year in the session – 2021-22)

The DTCs (Departmental Training Coordinators) would work in conjunction with the FPCs/DPCs (Faculty Placement Coordinators/Departmental Placement Coordinators) and the Head of the parent Department to plan and implement a special program for the students who have a special need to improve their English Communication skills to become industry ready.

An internal **pre-assessment** would be conducted (for QA + LR + VA + SST) at the start of the 5th semester according to the topics that will be taught in that semester, their **mid-assessment** would be conducted along with the PUE of 5th semester and their **post-assessment** would be conducted along with the PUE of 6th semester.

Like the QA + LR + VA pre-assessment being held at the start of the 5th semester, the **Soft Skills pre-assessment** would require all the students to submit their resume, for the **mid-assessment** along with the PUE of 5th semester, and for the **post-assessment** along with the PUE of the 6th semester.

For this pre-final year batch during both their semesters, Employability Skills training will be imparted – 20 hours of SST + 20 hours of QA & LR + 10 hours of VA.

The mandate of 60% compulsory attendance will be implemented strictly for all the students as per the Academic Policy of the Institute.

It will be tried to allocate one trainer to one batch for the whole year to get the desired outcomes, increase ownership, & accountability as well.

Students should be able to score at least 60% or more cut-off (in QA + LR & in VA) by the post assessment to become eligible for recruitment from campus.

Total number of training hours provided to all the students of this batch = 100

Final Year – All the students who register themselves with the CRPC Department and want to avail placement assistance from the campus will be provided capsule training according to their need.

NOTE-1: As the Quantitative Aptitude & Logical Reasoning sessions will not be conducted for the students of B. Pharmacy as a part of their regular timetable, so for them an AWS (Aptitude Winter School – 20 hours duration) will be conducted during the winter break to enhance their employability.

Topics index to be covered in each semester as per the above policy are attached herewith as appendices –

- Appendix A – B. Tech 1st year course structure – as per AKTU
- Appendix B – B. Tech 3rd semester
- Appendix C – B. Tech 4th semester
- Appendix D – B. Tech 5th semester & MCA 1st semester
- Appendix E – B. Tech 6th semester & MCA 2nd semester
- Appendix F – B. Pharma 5th, 6th, & 7th semester
- Appendix G – MBA – 1st, 2nd, & 3rd semester

Key points of implementing the training program:

- The respective HODs to accommodate Soft Skills & Aptitude Training – number of hours – in consultation with Head – H&S, within the regular timetable.
- An institute level literary event (prescribed under the extra-curricular activities calendar released from the Office of Dean-SW), **mandatory for all students**, comprising of three sub-events (debate, GD, & Group Presentation) would be conducted by the respective DTC (Departmental Training Coordinator) with the assistance of the parent department, and then the department level winners would compete to gain institute level recognition. The student clubs related to the H&S Department will also be involved in the organizing of this event.

This policy intends to be a guiding tool for the students at the KIET Group of Institutions and imbibe in them the need & importance of the training that they will go through to enhance their employability quotient. Our training and development initiative draws inspiration from the Vision of KIET Group of Institutions – 'to create competent professionals for the Industry and Society'.

Clubs related to TRAINING

E-boosters – E-boosters is a club under the aegis of the Department of Humanities and Social Sciences – KIET Group of Institutions, incepted in the year 2016. The sessions are conducted by the final year students for the second and third-year students to spread the strengths and alliance of knowledge to learn, enhance and grow in the skillset of English Communication, Aptitude and Logical Reasoning, Technical Aptitude and Coding. The peer-learning method in this club earnestly looks forward to the comprehended association with each-other to develop the professional skills required from competent global leaders with entrepreneurial orientation and a holistic approach to adapt to the rapidly changing world. To increase the ease and probability of gaining meaningful recruitment as per the current demand of the industries, the E-Boosters cadre with the able supervision of Soft Skills training team is uniquely operating within and beyond the prescribed curriculum. The chief intent is to emphasize the key subjects and push for the practice-based learning for applicable skill development and efficiency improvement of the students through inspirational and creative teaching that encourages students to assimilate, analyse and apply relevant knowledge. ***They will be actively involved in taking sessions beyond classroom hours, in the blended mode for the students to increase their proficiency in their low proficiency area(s).***

STC (Student Training Coordinators)–have created an online portal – CONNECTX (launched on 4th Aug'20) – to assist the students in identifying relevant job profiles & enhance their awareness about the recruitment process of the companies that hire from the KIET Group of Institutions. It is a student-driven teamwork project under the benign mentorship and guidance of the faculty members (Communication & Soft Skills) from the Department of H&S, in collaboration with the CRPC Department. This online endeavour of both the Departments has been initiated to facilitate the recruitment of students. The creation and maintenance of the portal was initiated by the final year students (2020-Batch), that will be carried forward by the next batch of students. The pre-final year students have also been proactive in the formulation of this learning platform. Governance is strictly maintained and coordinated under the vigilance of Faculty-Student Training Coordinators according to the guidelines enumerated in the club's SOP (standard operating procedure).

Policy formulated by: Mr Komal Mehrotra & Ms Puja Rohatgi
Policy approved by:
Policy creation date – 12th June 2019
Policy revised in – September 2021

Modifications suggested in: April 2020, by Mr Abhishek Gupta (Member Advisory Board)
May/June 2021 by Dr Manoj Goel (Joint Director – KIET), and in
July/August/September 2021 by Dr Amik Garg (Director – KIET)

The above policy to be implemented with immediate effect (from session 2K21-22), and it is understood to be dynamic in nature so modifications will continue to take place every semester/year as per the changes happening in the industry & online/offline nature of classes.

Mon - L1&L2

KIET Group of Institutions –Ghaziabad
Department of Humanities and Social Sciences

Sub: Quantitative Aptitude & Logical Reasoning
Trainer's Name: Manish Kr. Gupta
Branch/Sec: CSE - 3A
Semester: 3rd (Odd-Sem) **Session: 2022-23**

Session No.	Day/Date	Topic	Execution Date	Remark	Sign.
1	1st Sep to 9th Sep 22	Introduction and Discussion on Campus Placement & Recruitment Pattern, Speed Math	05/09/2022		Manish
2	12th Sep to 16th Sep 22	Number System-1(Number Classification, Prime Number, Co-Prime Number, Divisibility Rule, Factorials)	12/09/2022		Manish
3	19th Sep to 23rd Sep 22	Number System – 2(Unit Digit/Cyclicity, Remainder Theorem), **Pre-Assessment Test**	19/09/2022 26/09/2022		Manish
4	26th Sep to 30th Sep. 22	Percentage	17/10/2022		Manish
5	3rd Oct. to 7th Oct. 22	Profit Loss	21/11/22		Manish
6	18th Oct. to 22nd Oct. 22	Simple and Compound Interest	28/11/22		Manish
7	31st Oct to 4th Nov. 22	Series, Number Puzzle	09/01/23		Manish
8	7th Nov. 11th Nov. 22	Coding Decoding	14/11/2022		Manish
9	14th Nov. to 18th Nov. 22	Blood Relation	31/10/2022		Manish
10	21st Nov to 25th Nov. 22	DI (Tabular, Bar Graph, Pie chart, Line Graph) – Basic Level	09/01/23		Manish
11	28th Nov. 22 to 2nd Dec. 22	Non -Verbal Reasoning (Mirror Image, Water Image Non-verbal series)	16/01/23		Manish
12	5th Dec. 9th Dec. 22	Ranking & Order, Data Sufficiency	23/01/23		Manish
13	12th Dec. to 16th Dec. 22	Analytical Reasoning	5/12/22, 12/12/22		Manish
14	19th Dec. to 23rd Dec. 22	Back Up Week	30/01/23		Manish
15	26th Dec. to 30th Dec. 22	Problem Solving	03/10/2022, 02/01/23		Manish

- CT-1 Examination 10th to 17th Oct. 2022
- Deepawali Holiday 24th to 27th Oct. 2022
- ~~CT-2 Examination 21st to 26th Nov. 2022~~
- PUE Examination ~~2nd to 9th Jan 2022~~ 20th to 27th Feb 2023

CT-2 → 19th Dec - 24th Dec
class Suspended due to involvement in 3rd Year preplacement training
Semester Close – 11th feb 2023 ↳ 6th - 11th feb 2023

Thursday.
L5/L6
(To)
Tuesday
L1/L2

KIET Group of Institutions –Ghaziabad
Department of Humanities and Social Sciences

Sub: Quantitative Aptitude & Logical Reasoning
Trainer's Name: Vinod Agrawal.
Branch/Sec: EN/6B

Session:2022-23

Semester:6 th (Even Semester)

SN	Tentative Date	Topic	Execution Date	Remark	Sign.
1.	13 Feb- 17 Feb	Factors, LCM, HCF	16 Feb.		
2.	20 Feb- 24 Feb	Permutation, Combination & Probability	23 Feb.		
3	27 Feb- 03 Mar	Probability	14 March		
4	06 March- 10 March	Misc. Problem Solving			
5	13 March- 17 March	Data Arrangement	14 March		
6	27 March- 31 March	Syllogism	28 March		
7	03 April- 10 April	Critical Reasoning	04 April		
8	10 April- 14 April	Blood Relation, Dice	11 April		
9	17 April- 21 April	Capgemini, TCS NQT and other Mass Recruiters Test Pattern and Paper Discussion	18 April.		
10	24 April- 28 April	Back Up / Misc. Problem Solving	25 April		
		Problem Solving	02 may		

- CT-1 Examination: 20-25 March 2023
- Holi Holiday 7th to 8th March. 2023
- PUE Examination: May 1, 2023 Onwards

Tue - L(5&6)

KIET Group of Institutions –Ghaziabad

Department of Humanities and Social Sciences

Sub: Quantitative Aptitude & Logical Reasoning

Trainer's Name: Manish Kumar Gupta

Branch/Sec: ME-4A

Semester: 4th (Even Semester) **Session: 2022-23**

S No.	Day/Date	Topic	Execution Date	Remark	Sign.
1.	3rd April to 7th April	Ratio, Proportion and Variance	4/04/23	Done	Manish
2.	10th April to 14th April	Average, Mixture and Alligation	11/04/23 18/04/23	Average Mixture Alligation	Manish Manish
3.	17th April to 21st April	Syllogism, Possibility Case	25/04/23 02/05/23	Revision	Manish Manish
4.	24th April to 29th April	Time and Work, Efficiency based Question	16/5/23		Manish
5.	8th May to 12th May	Pipe and Cistern, Work and Wages	23/5/23		Manish
6.	15th May to 19th May	Clock and Calendar	30/5/23		Manish
7.	22nd May to 26th May	Direction, Cube and Dice			
8.	5th June to 9th June	Data Arrangement			
9.	12th June to 17th June	Data Interpretation			
10.	19th June to 23rd June	Problem Solving			

- CT-1 Examination: 1st May to 5th May, 2023
- CT-2 Examination: 29th May to 2nd June, 2023
- PUE Examination: 26th June to 30th June, 2023

MON (L3,L4)

KIET Group of Institutions –Ghaziabad
Department of Humanities and Social Sciences

Sub: Quantitative Aptitude & Logical Reasoning
Trainer's name: VINOD AGRAWAL
Branch/sec: CS /A
Semester: 5th (Odd sem) **Session:2022-23**

Session No.	Day/Date	Topic	Execution Date	Remark	Sign.
1.	29th Aug. to 2nd Sep. 22	Introduction and Discussion on Campus Placement & Recruitment Pattern.	29 Aug.	Done	
2.	5th Sep to 9th Sep 22	Speed Math, Number System-1(Number Classification, Prime Number, Co-Prime Number, Divisibility Rule, Factorials)	05 Sep.	"	
3	12th Sep to 16th Sep 22	Number System – 2(Unit Digit/Cyclicity, Remainder Theorem), **Pre-Assessment Test**	12 Sep.	"	
4	19th Sep to 23rd Sep 22	Percentage	19 Sep	"	
5	26th Sep to 30th Sep. 22	Profit Loss	26 Sep	"	
6	3rd Oct. to 7th Oct. 22	Simple and Compound Interest	03 Oct	"	
7	18th Oct. to 22nd Oct. 222	Number Series, Coding Decoding	31 Oct 07 Nov	" "	
8	31st Oct to 4th Nov. 22	~~Banking and Order,~~ Direction Sense, Data Arrangement.	21 Nov	"	
9	7th Nov. 11th Nov. 22	Analytical Reasoning			
10	14th Nov. to 18th Nov. 22	Back up week	14 Nov 21	"	

03 Oct: IMC Visit, Pragati Maidan

- CT-1 Examination 10th to 17th Oct. 2022
- Deepawali Holiday 24th to 27th Oct. 2022
- CT-2 Examination 21st to 26th Nov. 2022
- PUE Examination 2nd to 9th Jan 2022

CR - SHIKHA DIXIT - 9927476217. E-114, E-112 TUESDAY - 5&6.

KIET Group of Institutions, Ghaziabad

E-113.

Department of Humanities and Social Sciences

Sub: Soft skills | Trainer's name - SHRIKANT | Branch/sec: CS-C1

Semester: 6th (Even sem) | Session:2022-23

Session No.	Topic	Activity conducted	Scheduled Date	Executed Date	Remarks	Signature
1.	Recap & review on 5th sem (Resume feedback, E-portfolio/Video Resume	1-20 COUNTING GAME	13th to 17th Feb 2023	14/2/23	—	[signature]
2.	Discussion on present recruitment: online/offline and hybrid hiring. LinkedIn Profile Building (Networking)	EAGLE STORY	20th to 24th Feb. 2023	21/2/23	—	[signature]
3	Case-based GDs	—	27th Feb. to 3rd March 2023	28/2/23	—	[signature]
4	TMAY (elevator pitch)	—	6th to 10th March 2023	14/3/23	—	[signature]
5	Discussion on HR interview questions, pair interviews, and email writing	—	13th to 17th March 2023	28/3/23	—	[signature]
6	Peer Interviews	—	27th to 31st March 2023	11/4/23	—	[signature]
7	Group Discussion	—	3rd to 7th April 2023	4/4/23	—	[signature]
8	Email writing practice session	—	10th to 14th April 2023	18/4/23	—	[signature]
9	Presentations	—	17th to 21st April 2023	25/4/23	—	[signature]
10	Backup week	—	24th to 28th April 2023	—	—	—

Remarks:

CT 1- 20th to 25 March 2023

Holi Break 7th to 8th March 2023

PUE: 1st to 5th May 2023

* On 28th MARCH - COMBINED (C1 & C2) AS HS Sir on EXAM LEAVE MKH mam on AKTU duty.

* On 2nd May'23 - COMBINED BY HIMANSHU SIR - BUT - LATER INFORMED THAT NO CLASS DUE TO GUEST LECTURE.

ROOM NO: E-409; E 410(B)
FC:
CR: SHIKHA
9927476217

TUESDAY- 5th & 6th Lec.
CO-TRAINER- HS.

KIET Group of Institutions –Ghaziabad
Department of Humanities and Social Sciences

Sub: Soft skills
Semester: 5th (Odd sem)

Trainer's name - SHRIKANT KESHAV **Branch/sec:** CS-5C1
Session:2022-23

Session No.	Day/Date	Topic	Sessional Plan	Execution Date	Remark	Sign.
1.	29th Aug. to 2nd Sep. 22	Handing VUCA & BANI with power skills	Introduction Through ICE Breaker ABCDP	30th Aug'22	—	Sd
2.	5th Sep to 9th Sep 22	Becoming Career Ready	According to career choice group Presentation	6th Sep'22	—	Sd
3	12th Sep to 16th Sep 22	Future Skills & Changing Trends in Hiring Profile Understanding	Informational Interviewed Job Shadowing	13 Sep'22	—	Sd
4	19th Sep to 23rd Sep 22	(Personal Branding-1) CURE + Networking Linked In	Skill based hiring v/s Degree based hiring	20th Sept'22	—	Sd
5	26th Sep to 30th Sep. 22	Personal Branding-2	Resume/Video R & E-Portfolio	27th Sept'22	—	Sd
6	3rd Oct. to 7th Oct. 22	GD/CNA/PESTLE	Practice Session	18th OCT 22	RESUME FEEDBACK & INF INT-ASSIGN	Sd
7	18th Oct. to 22nd Oct. 222	Resume Feedback Info Int. Presentation	Generalized Individual	—	—	
8	31st Oct to 4th Nov. 22	Elevator Pitch (TMAY)	—	—	—	—
9	7th Nov. 11th Nov. 22	GD/JAM/Presentation	—	—	—	—
10	14th Nov. to 18th Nov. 22	Back up week	—	—	—	—

- CT-1 Examination 10th to 17th Oct. 2022
- Deepawali Holiday 24th to 27th Oct. 2022
- CT-2 Examination 21st to 26th Nov. 2022
- PUE Examination 2nd to 9th Jan 2022

* 31st oct - NO ACTIVITY AS CLASS WITHDRAWN.
* class suspended after 18th oct'22.
* 4th oct: RAMNAVMI - Holiday.

Mark

Time Table of TRAINING - Session 2022-23 (Even Semester)

B.Tech & B. Ph - 4th Semester

Day	I Period	II Period	III Period	IV Period	Lunch	V Period	VI Period	VII Period	VIII Period
	9:10-10:00	10:00-10:50	10:50-11:40	11:40-12:30	12:30-1:30	01:30-02:20	02:20-03:10	03:10-04:00	04:00-04:50
Monday	CSE-4A (QA & LR - MG) IT-4A(QA & LR - PC)				LUNCH	CS-4C(QA & LR - MG) CSIT-4C(QA & LR-PC)			
Tuesday	IT-4B(QA & LR - PC)		ME-4B(QA & LR-MG)			ME-4A(QA & LR-MG) CSIT-4B(QA & LR-PC)			
Wednesday	CSE-4B(QA & LR-MG)		IT-4C(QA & LR - PC)			CSIT-4A(QA & LR-PC)		CS-4A(QA & LR-MG)	
Thursday			CSE-4C(QA & LR-MG) CSE-4AI (QA & LR -PC)			CS-4B(QA & LR-MG) CE-4A (QA & LR - PC)			
Friday	ECE-4B(QA & LR-MG)		ECE-4A(QA & LR-MG) CSE-4AI & ML(QA & LR -PC)			EN-4A (QA & LR -PC) EN-4B (QA & LR-SPS)		ECE-4C(QA & LR-MG)	
Saturday									

PC - Mr Pawan Chaturvedi
MG - Mr Manish Gupta
SPS - Mr Sarvendra Pratap Singh

Mr Pawan Chaturvedi
Time Table Coordinator

Mr Komal Mehrotra
HOD-HSS

Time Table of TRAINING - Session 2022-23 (Even Semester)

B.Tech & B. Ph - 6th Semester

Day	I Period 9:10-10:00	II Period 10:00-10:50	III Period 10:50-11:40	IV Period 11:40-12:30	Lunch 12:30-1:30	V Period 01:30-02:20	VI Period 02:20-03:10	VII Period 03:10-04:00	VIII Period 04:00-04:50
Monday					LUNCH	ME-6A (QA&LR-ST) CE-6A (QA&LR-VA)		ECE-G1 (QA&LR -VA) ECE-G2 (QA&LR -SPS) ME-6B (QA&LR-ST)	
Tuesday			CS-6B (QA&LR-SPS) CSIT-6C (QA&LR-ST) EN-6B (QA&LR-VA)			CSE-6B (QA&LR-VA) IT-6A (QA&LR-ST)			
Wednesday	CSE-6C (QA&LR-VA) CSIT-6B (QA&LR-ST)					CS-6C (QA&LR-SPS) IT-6B (QA&LR-ST)			
Thursday						EN-6A (QA&LR-VA)		EN-6C (QA&LR-VA)	
Friday			CS-6A (QA&LR-SPS) CSIT-6A (QA&LR-ST)			CSE-6A (QA&LR-VA) IT-6C (QA&LR-ST)			
Saturday									

VA - Mr Vinod Agrawal
ST - Mr Shubham Tyagi
SPS - Mr Sarvendra Pratap Singh

Mr Pawan Chaturvedi
Time Table Coordinator

Mr Komal Mehrotra
HOD-HSS

Time Table of TRAINING - Session 2022-23 (Even Semester)

B.Tech & B. Ph - 6th Semester

Day	I Period 9:10-10:00	II Period 10:00-10:50	III Period 10:50-11:40	IV Period 11:40-12:30	Lunch 12:30-1:30	V Period 01:30-02:20	VI Period 02:20-03:10	VII Period 03:10-04:00	VIII Period 04:00-04:50
Monday			CSE-6A (KM+MKH) CS 6A (PR+HS) IT 6A (MK+HC) EN-AB (Placement only 4th Lec)		LUNCH				
Tuesday			CSE 6C (MK+MKH) CSIT 6A (KM+SP) ECE 6C (AM+IG) EN 6C (HS+AB)			CS 6C (SK+HS) IT6B (PR+HC) ECE 6A (SB+IG)			
Wednesday			CS 6B (SK+HS) IT 6C (MK+HC) EN-AB (Placement only 4th Lec)			ECE 6B (MKH+IG) EN-6A (HS+AB) ME 6A (PR+SB)			
Thursday	CE 6B(HS)		CSIT 6C (SK+SP) B.PH 6A (MKH+ SB)			B.PH 6B (HC+SB)			
Friday			CSIT 6B (SK+SP) EN 6B(KM+AB) ME 6B (MKH+SB)			CSE 6B (MK+MKH)		CE 6A (HS)	
Saturday									

KM - Mr Komal Mehrotra
PR - Ms Puja Rohatgi
HS - Mr Himanshu Saxena-SST
MK - Dr Meetu Kumar-SST
MKH - Ms Monika Khurana-SST & VA
SK - Mr Shrikant Keshav-SST & VA
HC - Mr Hitesh Chadha-SST

SP - Ms Sudesh Pathania-SST
IG - Ms Isha Grewal- SST
AB - Ms Ankita Banerjee - SST
SB - Ms Sugandha
AM - Ms Arunita Mukhopadhyay

Mr Pawan Chaturvedi
Time Table Coordinator

Mr Komal Mehrotra
HOD-HSS

Aptitude Pre Placement Training Lecture Plan (12 Hours)

6th Feb to 11th Feb 2k23

Day	Topics	Hours
1	Ratio Proportion, Mixture and Allegation	**2**
2	Work and Wages, Time and Work, Pipe and Cistern	**2**
3	Time Speed and Distance	**2**
4	Data Arrangement and Analytical Reasoning	**2**
5	Data Interpretation, Puzzles	**2**
6	Miscellaneous Problem Solving	**2**

Timetable mock Interviews for **ASSET - PRT-1**

1st Round-8th Dec 2k22 to 3rd Jan 2k23

2nd round- 16th May to 30th May 2K23

Resume collection during PUE:15th May 2K23

Aptitude tested during PUE:15th May 2k23

Technical Coding Test:25th May 2k23 (Programming Pathshala)

Pre-Placement Training (2k24 Batch) Phase-1 (Online) 10th July-14th August 2k23 (Ongoing)

Mock test Series

Aptitude Pre Placement Training Lecture Plan (12 Hours)

6th Feb to 11th Feb 2k23

Day	Topics	Hours
1	Ratio Proportion, Mixture and Allegation	**2**
2	Work and Wages, Time and Work, Pipe and Cistern	**2**
3	Time Speed and Distance	**2**
4	Data Arrangement and Analytical Reasoning	**2**
5	Data Interpretation, Puzzles	**2**
6	Miscellaneous Problem Solving	**2**

Time Table of TRAINING - Session 2022-23 (Odd Semester)

B.Tech - 1st, 3rd & 5th semester; MCA - 1st semester; MBA - 1st & 3rd semester; B. Ph - 5th

ay	I Period 9:10-10:00	II Period 10:00-10:50	III Period 10:50-11:40	IV Period 11:40-12:30	Lunch 12:30-1:30	V Period 01:30-02:20	VI Period 02:20-03:10	VII Period 03:10-04:00	VIII Period 04:00-04:50
nday			CSE-5A (KM+MKH) IT 5A (MK+HC) CS 5B (SK+HS)	CSE-5A (KM+MKH) IT 5A (MK+HC) CS 5B (SK+HS)		CSE 5B (MK+MKH) CS 5A (PR+HS) CSIT 5B (SK+SP) ECE 5B (KM+IG)	CSE 5B (MK+MKH) CS 5A (PR+HS) CSIT 5B (SK+SP) ECE 5B (KM+IG)		
esday			CSE 5C (MK+MKH) CSIT 5A (KM+SP) ECE 5C (AM+IG)	CSE 5C (MK+MKH) CSIT 5A (KM+SP) ECE 5C (AM+IG)	L	IT5B (PR+HC) CS 5C (SK+HS) ECE 5A (SB+IG) EN 5B(KM+AB)	IT5B (PR+HC) CS 5C (SK+HS) ECE 5A (SB+IG) EN 5B(KM+AB)	MCA 1C (HS+AM)	MCA 1C (HS+AM)
lnesday			IT 5C (MK+HC) CSIT 5C (SK+SP) CE 5A (HS) B.PH 5A (KM+ SB) MBA 3B (PR+MKH)	IT 5C (MK+HC) CSIT 5C (SK+SP) CE 5A (HS) B.PH 5A (KM+ SB) MBA 3B (PR+MKH)	U N	EN 5C (HS+AB) ME 5B (MKH+SB)	EN 5C (HS+AB) ME 5B (MKH+SB)	MCA 1B (MKH+AM)	MCA 1B (MKH+AM)
ursday			EN-5A (HS+AB) ME 5A (PR+SB) MBA 3C (IG+MKH)	EN-5A (HS+AB) ME 5A (PR+SB) MBA 3C (IG+MKH)	C H	CE 5B (HS) MBA 1A (MKH+SP)	CE 5B (HS) MBA 1A (MKH+SP)		
Friday	MBA 1C- (MKH+AB) MBA 1B (HS+AM)	MBA 1C- (MKH+AB) MBA 1B (HS+AM)	B.PH 5B (HC+SB) MBA 3A (PR+MKH)	B.PH 5B (HC+SB) MBA 3A (PR+MKH)		MCA 1 A (PR+AM)	MCA 1 A (PR+AM)		
aturday									

KM - Mr Komal Mehrotra		SP - Ms Sudesh Pathania-SST
PR - Ms Puja Rohatgi		IG - Ms Isha Grewal- SST
HS - Mr Himanshu Saxena-SST		AB - Ms Ankita Banerjee - SST
MK - Dr Meetu Kumar-SST		SB - Ms Sugandha
MKH - Ms Monika Khurana-SST & VA		AM - Ms Arunita Mukhopadhyay
SK - Mr Shrikant Keshav-SST & VA		
HC - Mr Hitesh Chadha-SST		

Time Table of Aptitude TRAINING - Session 2022-23 (Odd Semester)

B.Tech - 3rd & 5th semester; MCA - 1st semester; MBA - 1st & 3rd semester

I Period	II Period	III Period	IV Period	Lunch	V Period	VI Period	VII Period	VIII Period
9:10-10:00	10:00-10:50	10:50-11:40	11:40-12:30	12:30-1:30	01:30-02:20	02:20-03:10	03:10-04:00	04:00-04:50
CSE-3A (QA&LR-MG) CSIT-3A (QA&LR-PC) MCA-1A (QA&LR-SPS		EN-3A (QA&LR-PC) CS-5A (QA&LR-VA) IT-5B (QA&LR-ST)			ECE-3C (QA&LR-MG) ECE-5A (QA&LR -VA) IT-5C (QA&LR-ST)		IT-3A (QA&LR-PC) MBA-1C (QA&LR-SPS)	
CSE-3B (QA&LR-MG) CSIT-3B (QA&LR-PC) IT-5A (QA&LR-ST)		EN-3B (QA&LR-SPS) CS-5B (QA&LR-VA)			ME-3A (QA&LR-MG) ECE-5B (QA&LR -VA) CSIT-5B (QA&LR-ST)		IT-3B (QA&LR-PC) MCA-IB(QA&LR-SPS)	
CSE-3C (QA&LR-MG) CSIT-3C (QA&LR-PC) CSE-5A (QA&LR-VA)		CS-5C (QA&LR-VA) EN-5A (QA&LR-SPS)			MCA-1C (QA&LR-SPS) ME-3B (QA&LR-MG) ECE-5C (QA&LR-VA) ME-5A (QA&LR-ST)		IT-3C (QA&LR-PC)	
ECE-3A (QA&LR-MG) MBA-1A (QA&LR-SPS) CSIT-5C (QA&LR-ST)		ECE-3B (QA&LR-MG) MBA-3A (QA&LR-VA) EN-5C (QA&LR-SPS)			CSML-3A (QA&LR-PC) MBA-3B (QA&LR-SPS) EN-5B (QA&LR-ST)		CS-3B (QA&LR-MG)	
CS-3A (QA&LR- MG) CSE-5B (QA&LR-VA)		CSAI-3A (QA&LR-PC) CSE-5C (QA&LR-VA) CE-5A (QA&LR-ST) MBA-1B (QA&LR-SPS)			CE-3A (QA&LR-PC) MBA-3C (QA&LR-SPS) ME-5B (QA&LR-ST)		CS-3C (QA&LR-MG) CSIT-5A (QA&LR-ST)	

VA - Mr Vinod Agrawal
ST - Mr Shubham Tyagi

PC - Mr Pawan Chaturvedi - QA & LR
MG - Mr Manish Gupta - QA & LR
SPS - Mr Sarvendra Pratap Singh

Annexure to Office Order 27/2020 dated 18th Nov'20

Policy Document - Skill Development & Finishing School (SD&FS)

Objective: The changing demands of the Industry require more emphasis on coding as a culture, industry-based project and soft skills. The Institute is already working on these aspects in a distributed manner but *coding as a culture* in core engineering branches is not properly addressed. So, a separate vertical **named Skill Development & Finishing School (SD&FS)** is created to address these industry demands and align these distributed activities.

The **structure of the vertical** is shown in *Figure 1*. It is shown below

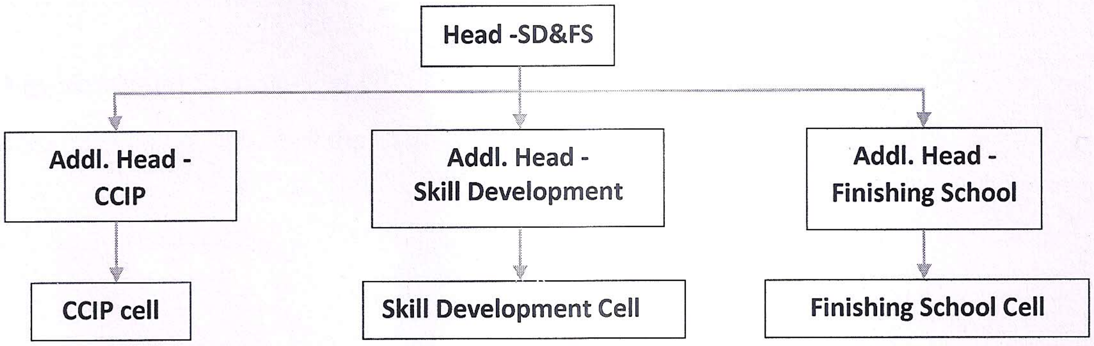


Figure 1: Organizational structure of SD&FS

1. **Mr. Prashant Agrawal, Associate Professor-MCA** appointed as **Addl. Head- CCIP.** This appointment is additional to his regular duties. He will be responsible for planning and smooth conduction of **competitive coding improvement program (CCIP)** for core branches through respective Departments. He will report to Head -SD&FS for CCIP cell. The detailed description is attached at **Appendix-1**.

2. **Dr Vipin Kumar, Associate Professor-MCA** redesignated as **Addl. Head- Skill Development**. This appointment is additional to his regular duties as Associate Professor. He will be responsible for planning and smooth conduct of various skill development activities including Minor Specialization & various technical clubs/Centre of excellences. He will report to Head-SD&FS for this responsibility. The detailed description is attached at **Appendix-2.**

3. **Ms Puja Rohatgi, Addl. Head-HS** is appointed as **Addl. Head- Finishing School.** This appointment is additional to her regular duties. She will be responsible for planning and execution of finishing school. She will report to Head-SD&FS for this responsibility. The detailed description is attached at **Appendix 3.**

02 Dec 20

APPENDIX-1

Policy Document for CCIP Module

CCIP- Competitive Coding Improvement Program, is a new initiative of KIET Group of Institutions for fulfilling the need of current trends in hiring freshers in big IT companies. Nowadays all IT companies are focusing very much on coding skills as well as other skills of students.

This document will give you a brief about **organizational structure** and **working process** of the **CCIP module.**

1.0 Organizational Structure of CCIP Module

The organizational structure of the CCIP module is depicted in Figure 2.0.

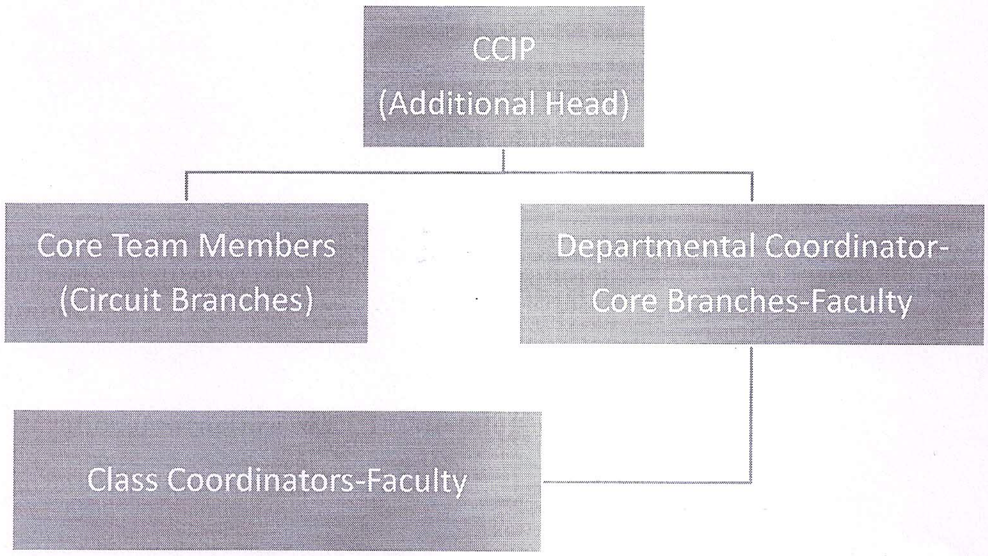


Figure 2: Organizational Structure of CCIP Module

02 Dec 20

2.0 Hierarchical Structure of CCIP module

The hierarchical structure of the CCIP module is shown in Figure 3.

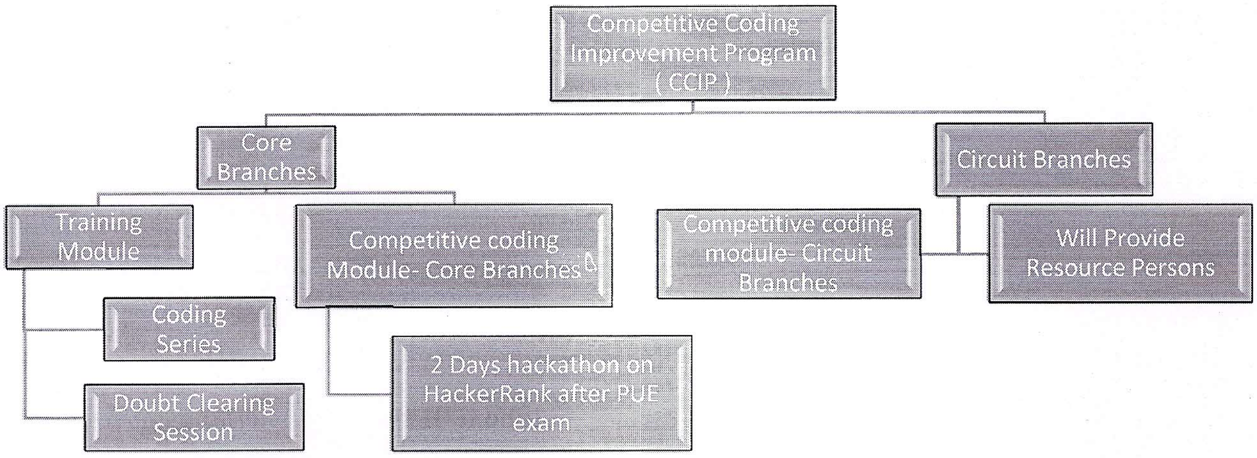


Figure 3: Hierarchical Structure of CCIP Module

3.0 Process of Training Module

The process of training module is shown in Figure 4.

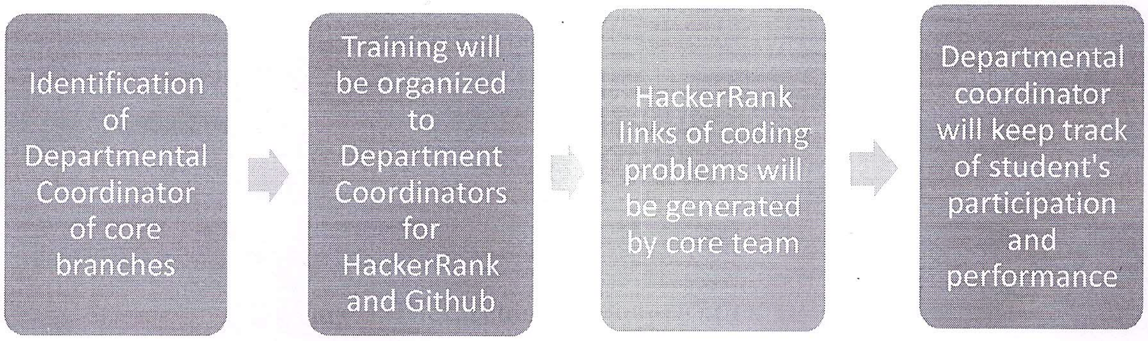


Figure 4: Process of Training Module

02 Dec 20

4.0 Role and Responsibility of members - CCIP Module

4. 1 Role of Resource Person

- Creating and uploading question set on HackerRank
- Providing technical guidance to students during doubt clearing sessions
- Provide technical training to departmental coordinator regarding HackerRank/GitHub/LinkedIn

4.2 Role of Respective Department

- Department will be responsible to record attendance and maintain discipline of the doubt clearing session.
- The Department will ensure that the students submit their problem solutions on HackerRank, share their code on GitHub and share their learning on LinkedIn.
- The Department will motivate and ensure the 100% participation of students in coding series as well as doubt clearing session.

5.0 Appreciation to students as per their involvement

5.1 Training Module

- Consistency certificates for 100 % participation in throughout the semester
- Star performer certificates for performing better than average
- Achiever Certificates for performing best in throughout the semester
- All certificates will be issued department and semester wise

5. 2 Competitive Module

- Top Three students will be awarded by certificates and cash prize at institute level
- Top Three students will be awarded by certificates and cash prize at department level

CCIP for Session 20-21

- Introductory classes on Python will be organized by resource persons for core branches for second year students
- From 4th and 6th semester same CCIP modules can be introduced to all students

Modus Operandi

- Coding Problems will be published on HackerRank per week.
- Students should submit the problems on Hacker Rank within given time period.
- Two lectures per week will be scheduled for doubt clearing session for the problems published in previous week.
- Students will get the regular attendance for attending the doubt clearing session.

02 Dec 20.

APPENDIX-2

Policy for Skill Development

Administrative Structure

The administrative structure of the Skill Development Cell is shown in Figure 5.

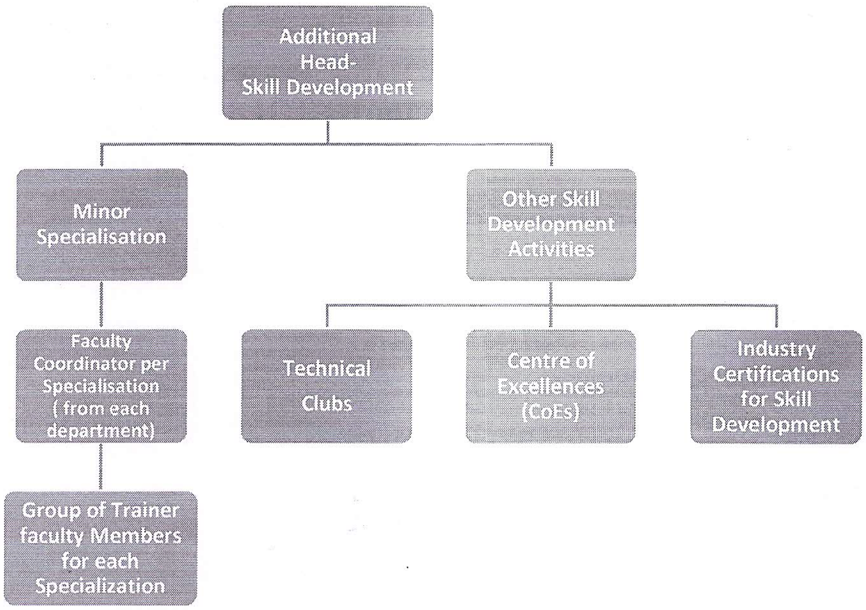


Figure 5: Administrative Structure of Skill Development Cell

Minor Specialization: -

1. Each department should select their own minor specialization area as per their faculty strength and based on industry requirements and student's needs.
2. These minor specialization courses will be offered for interested student of all departments.
3. Minor specialization will be offered in summer break.
4. Evaluation work cum Certificate distribution will be performed in 6th semester.
5. MOOC courses should be offered to students during semester.
6. During semester, all students should work in Excellence Centers (CoEs), Department and Innovation Clubs as per their selected minor specialization.

02 Dec 20

7. Minor specialization course duration will be 6 weeks (180 hrs.) in summer break. The suggestive distribution will be:
    a. 45 hrs. Training
    b. 45 hrs. Practical
    c. 90 hrs. Project Work
8. Departments will appoint a faculty coordinator for coordinating the minor specialization course.
9. Respective Department will be fully responsible for successfully running their minor specialization course.
10. Each department will design their own syllabus for their minor specialization and this syllabus should be approved from Board of Studies (BoS) of respective Department.
11. Syllabus should be practical oriented and based on industry requirements.
12. Syllabus should be planned for two parts. First part should be basic or fundamental that should be offered in 1st summer break and 2nd part should be advanced that should be offered in 2nd summer break.
13. Department should train or arrange their faculty member as per their selected minor specialization area.
14. Inter-Department expert faculty members group should be formed for teaching the respective minor specialization.

Modus operandi:

1. Finalization of minor specialization areas
2. Finalization of Faculty Coordinator from each department
3. Finalization of formation of Inter-Departmental expert faculty members group
4. Finalization of Syllabus & MOOC Courses (if possible) and its approval from respective Department Board of Studies (BoS).
5. Timetable for summer break

Other Skill Development Activities:

- All responsibilities w.r.t. Technical Clubs, CoE & Industry Certifications e.g Coursera / faculty skill development initiatives etc. for Skill Development will continue to be performed as assigned under the previous appointment of Asst. Dean Academics (Skill Development).

02 Dec 20

Appendix-3

Finishing School Policy

The Finishing School at KIET Group of Institutions has been created with a view to provide final polishing touches to the students across all domains just before appearing for their recruitment process and getting hired as a professional in the corporate world after earning **the ASSET (Aptitude + Soft Skills + English + Technical Proficiency) Certificate** within a period of 4 – 8 weeks. The number of hours invested during this program will be counted as **a part of the students' internship**, and hence considered to be mandatory.

As soon as the student ends the pre-final year exams, he/she will undergo a mix of external third-party training and an in-house training, to meet the following objectives –

- To groom students as per the company standards
- To hone the skills of the students in three verticals viz., Aptitude (QA + LR + VA), Soft Skills, and Technical Skills in tandem with the current industry needs
- To prepare the students for Recruitment for mass recruiters
- To provide students with the "ASSET" certificate for readiness towards recruitment drives

As a pilot program, this will be starting from the 2021 passing out batch of B. Tech and MCA. Modalities for Pharmacy & MBA students shall follow separately.

Modus operandi …

- All students will undergo a 17-days training that will be imparted to them as follows –
    - A 2-day input on Soft Skills – tips on Resume Writing, Group Discussions, and Personal Interviews from the faculty of H&S Department – Training Wing.
    - A 10-day input on Aptitude (QA + LR + VA) / Cognitive Assessment by an external 3rd party vendor.
    - A 5-day input on Technical topics useful for recruitment tests by an Industry expert, Alumni, or an Expert Faculty from our Institute. The two days will be given to problem solving and coding skills and remaining three days will be given to branch specific technical training. Respective Departments will be responsible to conduct the branch specific technical training for three days.

- On completion of this training, all the students will appear for **the first placement readiness test (PRT-1)**, based on the pattern of the major mass recruiters that will have sectional cut-off, so as to identify the problem area of particular students.

- Those students who qualify the PRT-1 will appear for a mock panel interview round with a technical resource person (Industry expert/Alumni/Institute Faculty) and an in-house HR expert / Soft Skills Trainer (number of days required may vary according to the

02 Dec 20

number of students that qualify PRT-1), will be conducted parallel to the refresher being run for the students who fail to qualify PRT-1.

- Those students who qualify the interview will gain the ASSET Certification.
- Those students who are unable to qualify the interview process will get one-on-one mentoring sessions with a Soft Skills Trainer.

➤ Those students who are unable to qualify PRT-1, will get a one-week refresher/doubt removal session that will be conducted by in-house resources on their domain of improvement and then **appear for PRT-2** that will be conducted on similar lines to PRT-1.

➤ Those students who qualify the PRT-2, will appear for a mock panel interview (technical + HR – here the resource persons would be internal) .

- Students who qualify the interview will gain the ASSET Certification.
- Students who are unable to qualify the interview process will get one-on-one mentoring sessions with a Soft Skills Trainer.

➤ Those students who are unable to qualify PRT-2, will again undergo a one-week refresher/doubt removal session according to their improvement area, and then **appear for a final PRT-3** based on similar lines as PRT-1.

➤ Those students who qualify the PRT-3, will appear for a mock panel interview (technical + HR)

- Students who qualify the interview process will gain the ASSET Certification.
- Students who are unable to qualify the interview process will not be awarded the ASSET Certification.

Modus operandi for the SCHOLAR BATCH

➤ Those students who **qualify PRT-1 with an above average score** in all the sections and also **perform well in the interview process** will be considered as the premium/scholar batch of students who will be eligible for better job profiles/packages.

➤ These students will be imparted special training for the remaining duration of the summer break to prepare them for the better profiles/packages (resource persons from the industry / alumni / faculty to be used as per requirement).

02 Dec 20

Modus operandi for Capsule Trainings of selected companies

- After receiving the relevant information regarding the next recruitment drive from the CRPC Department, and taking the number of days available for the recruitment process to commence, a company specific training would be organized as per its recruitment pattern.
  - The duration of the training and the requirement of resource persons for the training would be dependent on two variables – first, is the number of days available, and second is the number of students eligible/registered for the process.

Administrative Structure of the Finishing School:

The administrative structure of the finishing school is shown in Figure 6.

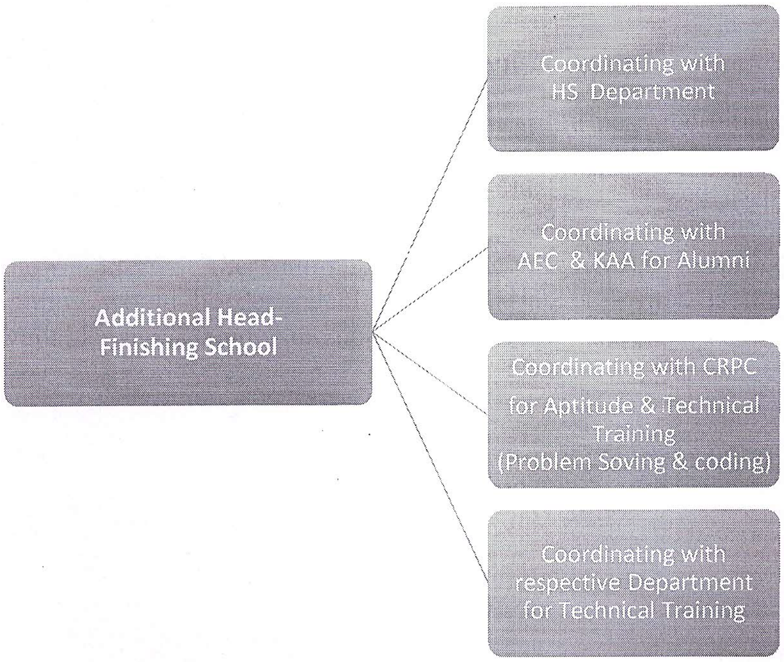


Figure 6: Administrative Structure of Finishing School

Note:

1. **Student Training Coordinators (STCs)** - One student (final year)/section would be assigned as Student Training Coordinators (STCs) who will perform their role under the overall directions of Addl. Head- FS as per requirement projected from time to time.
2. **Nomination of SPOCs - Soft Skill & Technical Skill Training** – In case SPOCs are needed from concerned dept., Addl. Head-FS will nominate in consultation with Head-SD&FS.

02 04 20

Semester wise detailed Syllabus for CCIP Training Modules

B.Tech 3 Sem (Programming Concepts)

week	course	Topics
1	operators	types of operators (Arithmetic, Relational, Logical, bitwise)
2	conditional statements	jump statements, conditional operator, if-else, switch case
3	loops	nested loops & problems on pattern printing
4	arrays 1-D	initialization, insertion and traversing array
5	Array 2-D	initialization, insertion, problems on matrix
6	strings	initializing string & in- built features
7	functions	function declaration & calling, variable arguments
8	Pointers	concept of pointers, array & pointers, functions & pointers
9	Dynamic memory allocation	C: malloc, calloc

B.Tech 4 Sem (Data Structures)

week	course	Topics
1	Data Structures basics	basics of DS, Algorithm analysis & structures
2	array	1-D array, 2-D array
3	Recursion	Recursive method calls, pointers
4	stacks	implementing stack using array, push & pop operations
5	queue	implementing queue using array, enque & deque operations
6	linked list	concept, creating & inserting nodes
7	linked list	implementing stack & queues
8	Sorting	Insertion Sort, Selection Sort
9	Sorting	Bubble sort, heap sort

B.Tech 5 Sem (Data Structures)

week	course	Topics
1	Data Structures Overview	
2	Graphs	DFS, BFS
3	Searching	Liner Search, Binary Search, concept of hashing
4	Trees	Binary tree: insertion & deletion operations
5	Trees	traversals
6	Divide & conquer	Merge Sort
7	Divide & conquer	Quick Sort
8	Greedy Programming	concept & Applications
9	Dynamic Programming	concept & Applications

B.Tech 6 Sem (Data Structures)

week	course	Topics
1	Greedy Programming	knapsack
2	Greedy Programming	Huffman Coding
3	Dynamic Programming	Shortest Path
4	Dynamic Programming	Longest Common subsequences
5	pseudo codes 1	Company Based problems
6	pseudo codes 2	Company Based problems
7	pseudo codes 3	Company Based problems
8	pseudo codes 4	Company Based problems
9	pseudo codes 5	Company Based problems

Prashant Agrawal
Addl Head CCIP

Dr. Ajay K. Shrivastava
Head-SD&FS

KIET GROUP OF INSTITUTIONS, DELHI-NCR, GZB

Department of Skill Development & Finishing School-CCIP

Time Table of CCIP - Session 2022-23 (Even Semester) **w.e.f 3.04.23**

Day	I Period	II Period	III Period	IV Period	Lunch	V Period	VI Period	VII Period	VIII Period
	9:00-09:50	09:50-10:40	10:50-11:40	11:40-12:30		01:20-02:10	02:10-03:00	03:10-04:00	04:00-04:50
Monday								ME-IVA:KOM	
Tuesday								EC-IV: AKU,KOM,SHW	
Wednesday									
Thursday			CE-IVA:KOM					EN-IV-A:AKU EC-VI: KOM,SHW	
Friday								ME-IVB:AKU EN-IV-B: KOM	

AKU : Dr. Amit Kumar
SHW: Ms. Shweta Singh
KOM: Ms. Komal Salgotra

EC department will run CCIP classes in Gradual and Galent Group
EN,CE and ME will run section wise

Mr. Prashant Agrawal
Addl Head - CCIP

Dr. Ajay Kr. Shrivastava
(Head-SD&FS)

KIET GROUP OF INSTITUTIONS, DELHI-NCR, GZB

Department of Skill Development & Finishing School-CCIP

Time Table of CCIP - Session 2022-23 (Odd Semester)

w.e.f 17.10.22

Day	I Period	II Period	III Period	IV Period	Lunch	V Period	VI Period	VII Period	VIII Period
	9:00-09:50	09:50-10:40	10:50-11:40	11:40-12:30		01:20-02:10	02:10-03:00	03:10-04:00	04:00-04:50
Monday						ME IIIA- AKU		EN III A - SHW	
Tuesday								EN IIIB-KOM EC VC-AKU	
Wednesday			ME VA- SHW			EC VA- SHW		EN VA-AKU EC IIIB-KOM	
Thursday			ME VB-KOM CE IIIA-AKU			EC VB-KOM		EC IIIA - SHW EN VB-AKU	
Friday	CE VA-SHW					ME IIIB-SHW EC IIIC-AKU		EN VC-KOM	

AKU : Dr. Amit Kumar
SHW: Ms. Shweta Singh
KOM: Ms. Komal Salgotra

Mr. Prashant Agrawal
Addl Head - CCIP

Dr. Ajay Kr. Shrivastava
(Head-SD&FS)

KIET GROUP OF INSTITUTIONS
CPP CELL FIRST YEAR TENTATIVE SYLLABUS
(Computer Programming Proficiency Cell)
Levels of Programming Challenges

Practice – To be attempted by students only.
L1 (Beginners) – Few challenges to be solved by Faculty and TA
L2 (Intermediate) – Few challenges to be solved by Faculty, TA and STA
L3 (Advanced) – To be attempted by students only.

***Note**: Atleast eight to ten programming challenges for each level need to be designed for each topic along with the test cases for automated grading on HackerRank.*

Module 1 (Odd Semester)

1. Basic Programming Techniques (input/output, loops, conditional statements, functions etc.) and Complexity Analysis (space and time restrictions on output) [Prof Alok]
2. Arrays (Kth maximum and minimum number, rearrangement, cyclic shift, subsequences etc.) and Matrices (rotations, inversions, chain multiplication, submatrices etc.) [Prof Shalika]
3. Searching and Sorting [Prof Ritu]
4. Number Based Problems: - Different Types of Numbers (Armstrong, perfect, automorphic, special etc.), Combinatorics, Number Theory (primality testing, exponentiation, modular arithmetic etc.), Probability, Bit Manipulation [Prof Anubhav and Prof Naveen]
5. Recursion and Backtracking (gcd, array reversal, recursive binary search etc.) [Prof Hriday]
6. String Manipulation (pattern search, substring removal, simple encryption, pattern based splitting etc.) [Prof Rajiv]

Module 2 (Even Semester)

To be decided (C++/Java, Standard Template Library (STL), basic data structures and algorithms)

Reference

1. Laaksonen, Guide to Competitive Programming, 2nd ed., Springer International Publishing, 2020.
2. Johan Sannemo, Principles of Algorithmic Problem Solving. Draft version. 2018
3. S. Skiena and M. Revilla, Programming challenges. New York: Springer, 2005.

Suggestions for Designing Programming Challenges

- Consult practice problems on HackerRank, HackerEarth and Codechef to understand how to frame programing challenges, put space/time restrictions, and design test cases
- Outcome of solving a programming challenge can be associated with them
- Try to ensure that the programming challenges are not available verbatim on Google. The language can be changed for existing problems
- Level 1 problems must be easy enough to motivate students, level 2 problems must be difficult enough that a student need to brainstorm for the solution, and level 3 problems must be hard enough that students will need to spend considerable amount of time exploring the best strategy to solve them
- Try to draft at least a few novel programming challenges for level 2 and 3

Student CPP Manual

Department of Computer Science and Engineering

Table of contents

Objectives of CPP Program

Guidelines to create Hackerrank Profile

1. *Open* URL: www.hackerrank.com
2. *Click* on **Sign Up & Code** (*for Job Seekers*) as shown in Figure 1.

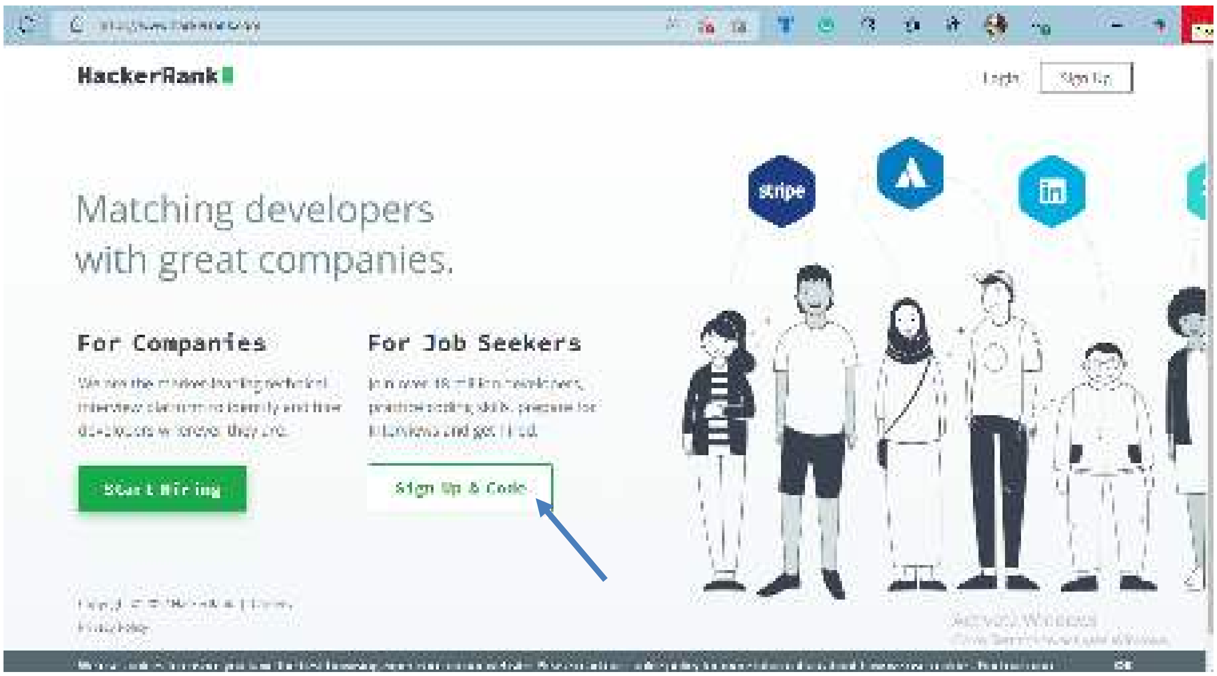


Figure 1. Hackerrank Home page

3. Fill the necessary details required to create an account and *click* on **Create An Account** as shown in Figure 2.

Figure 2. Sign Up page

4. Once sign-up is done, *profile personalization page* appears, *click* on **Learn & Compete with Others** as shown in Figure 3.

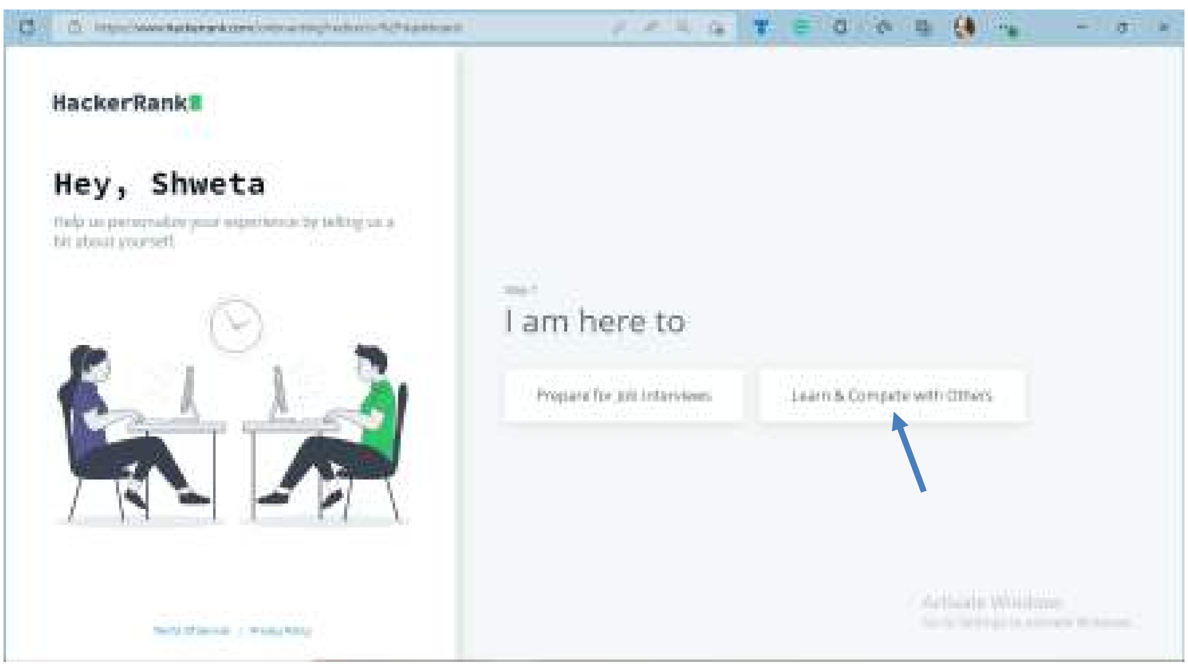


Figure 3. Step-1 (Profile personalization page)

5. *Choose* an appropriate category as shown in Figure 4.

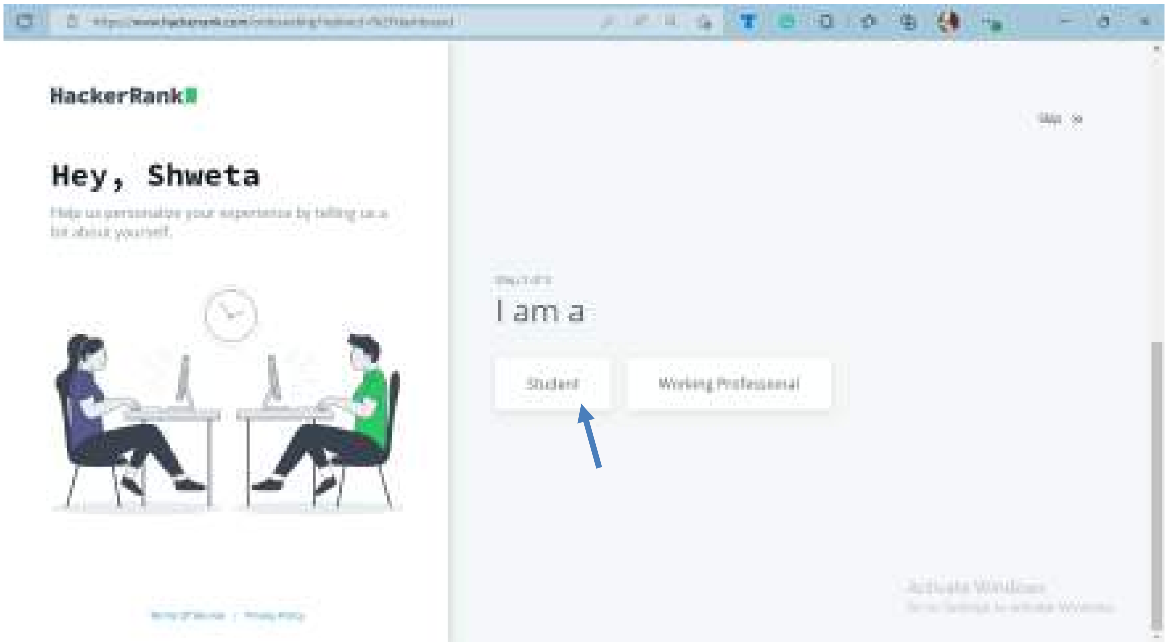


Figure 4. Step-2 (Profile personalization page)

6. *Choose* an year in which you will be graduating, *check the below checkbox* if you feel interested in being contacted through Hackerrank about various job opportunities and *click* on **Let's go** button as shown in Figure 5.

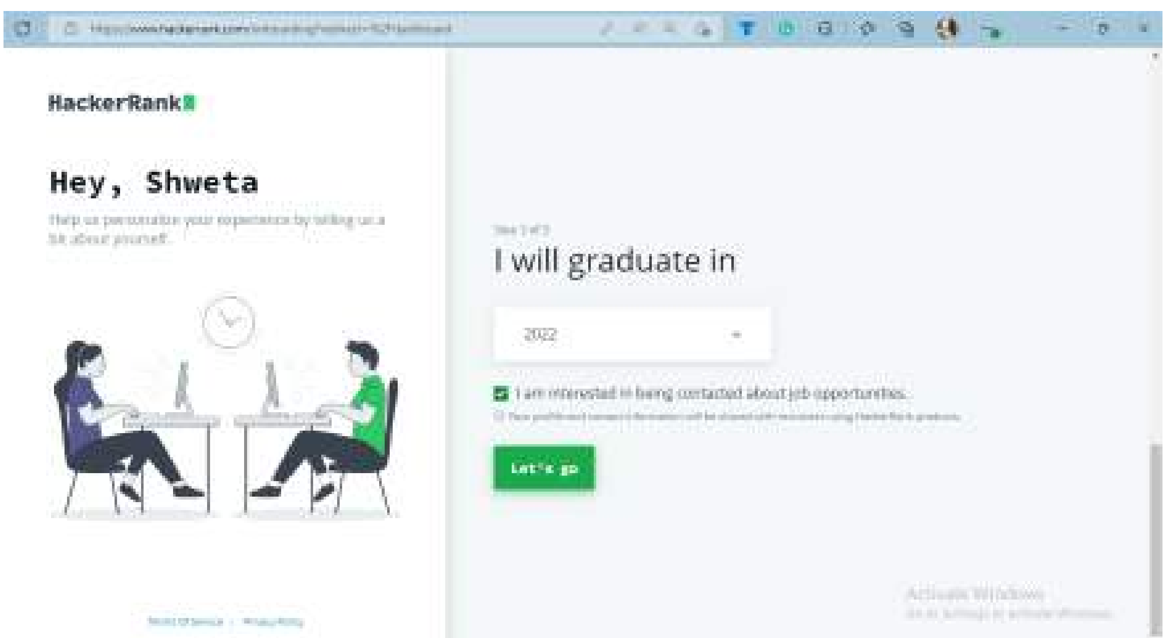


Figure 5. Step-3 (Profile personalization page)

7. Once, all three steps are completed, your profile **Dashboard** will appear as shown in Figure 6, *confirm your email address* by visiting your email account and *clicking* on **Confirm** as shown in Figure 7.

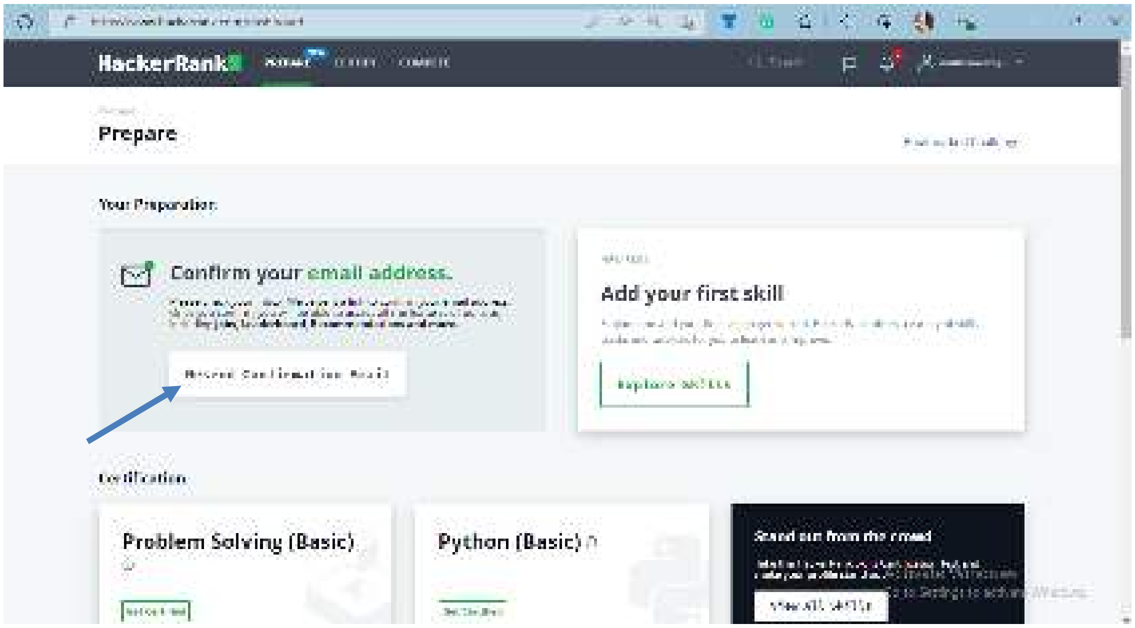


Figure 6. Profile Dashboard

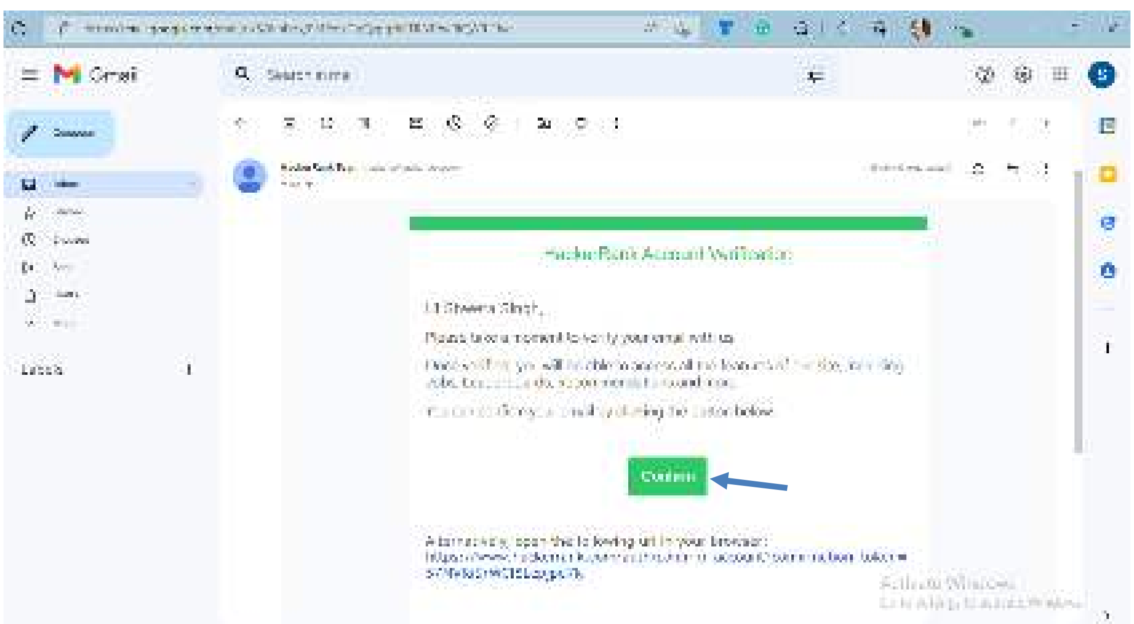


Figure 7. Email Confirmation page on gmail.com

8. By confirming your email address, you will be *directed* to the **Hackerrank Dashboard** page as shown in Figure 8.

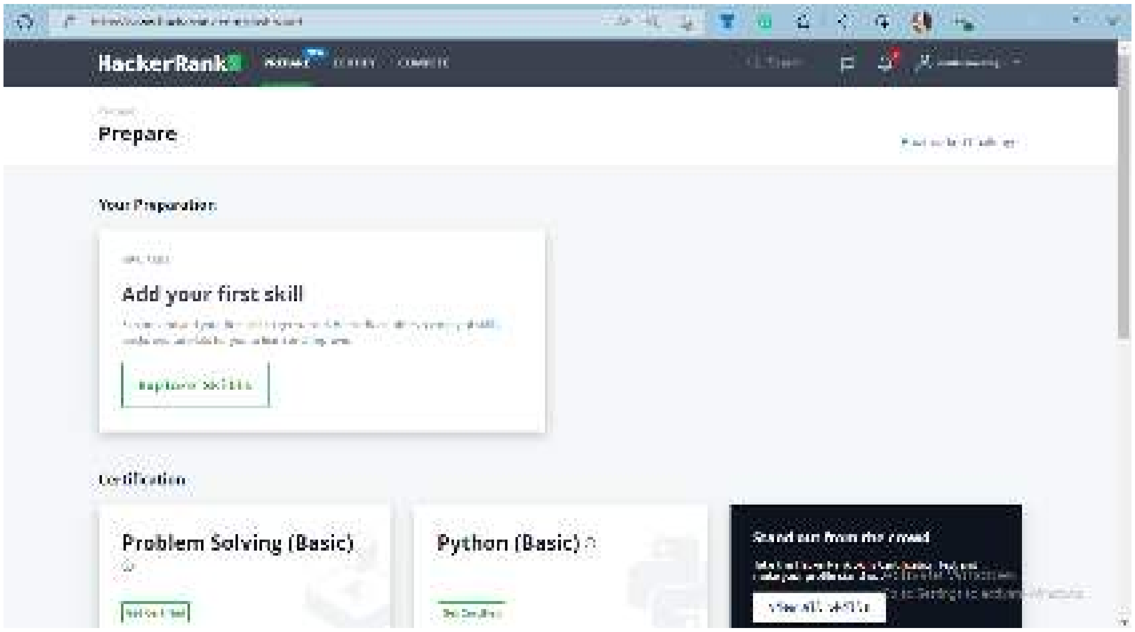


Figure 8. Hackerrank Dashboard page

9. To complete the profile, *click* on the arrow beside your username and *select* **Profile** from the drop-down menu to follow the steps of profile completion as shown in Figure 9, 10 and 11.

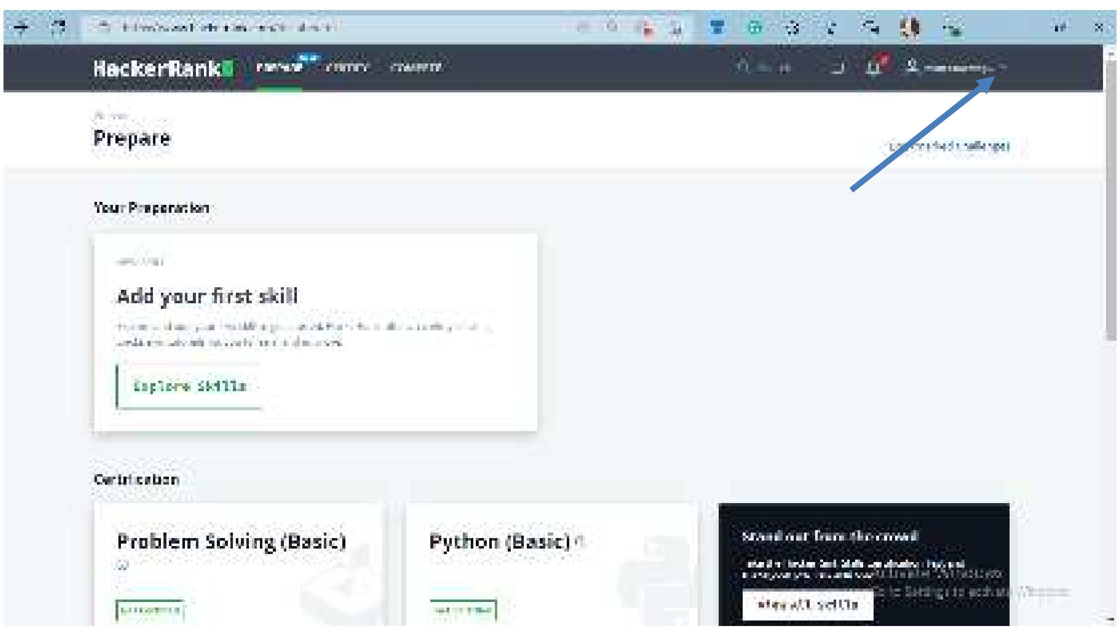


Figure 9. Step-1 Profile Completion

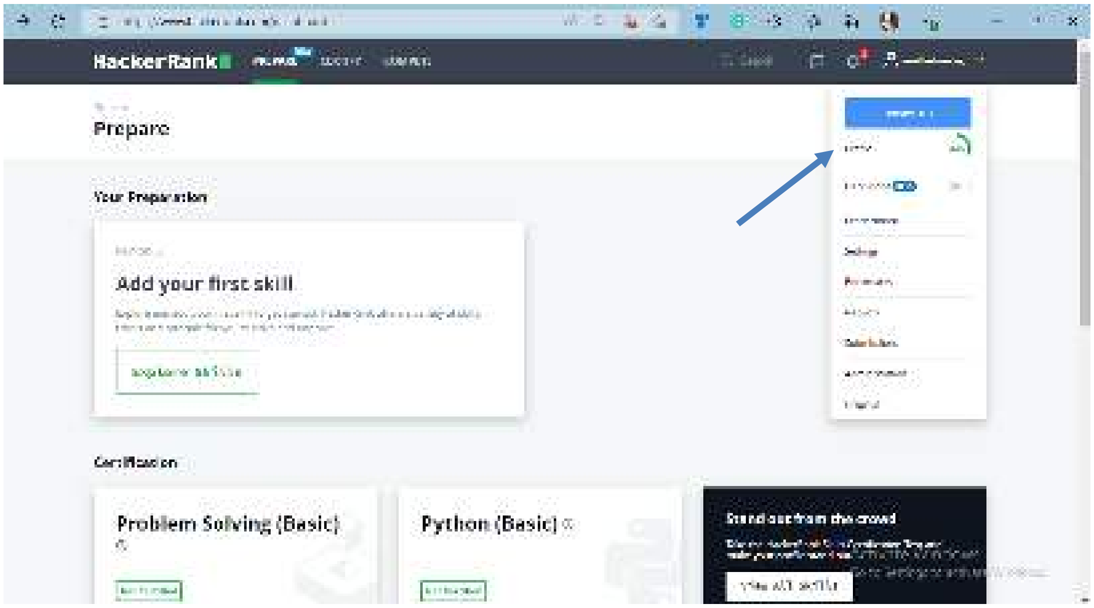


Figure 10. Step-2 Profile Completion

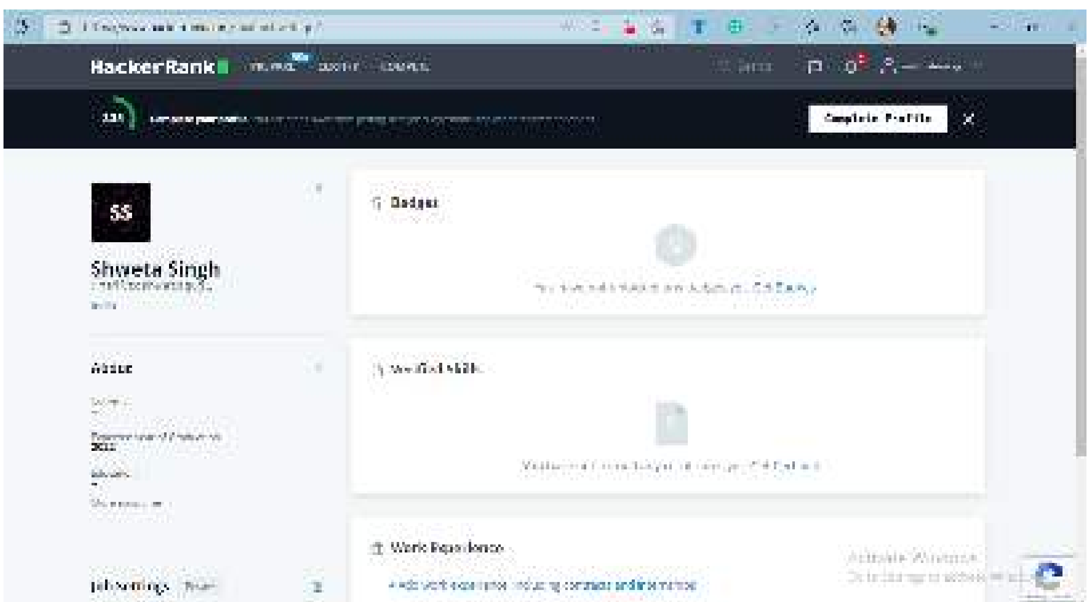


Figure 11. Step-3 Profile Completion

10. At last, to *change your username* and *connect via various social platforms*, *select* **Settings** from the same drop-down menu (as shown in Figure 9) and *update the necessary links* as shown in Figure 12 and 13.

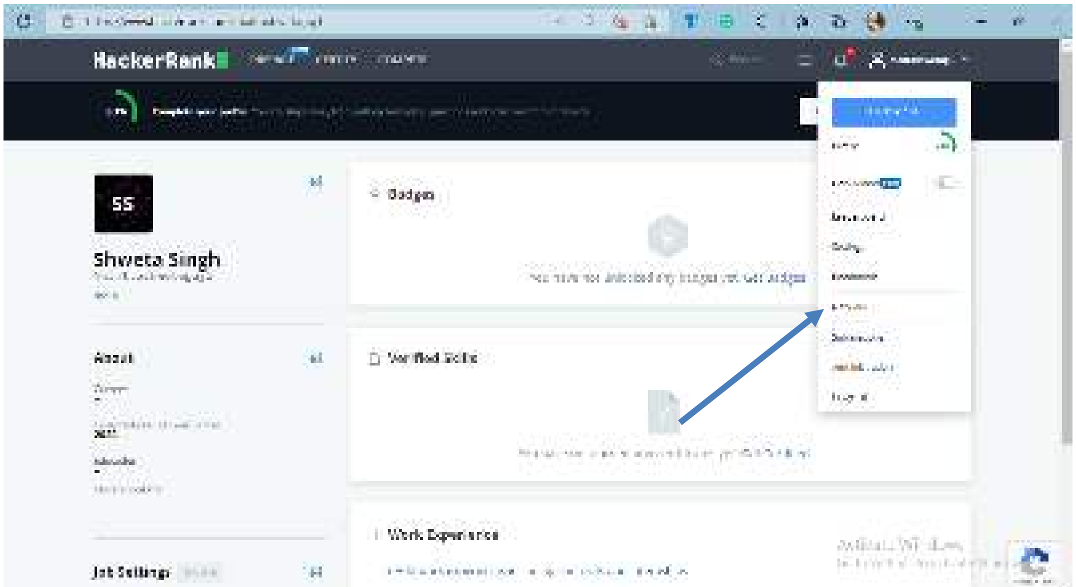


Figure 12. Settings Menu

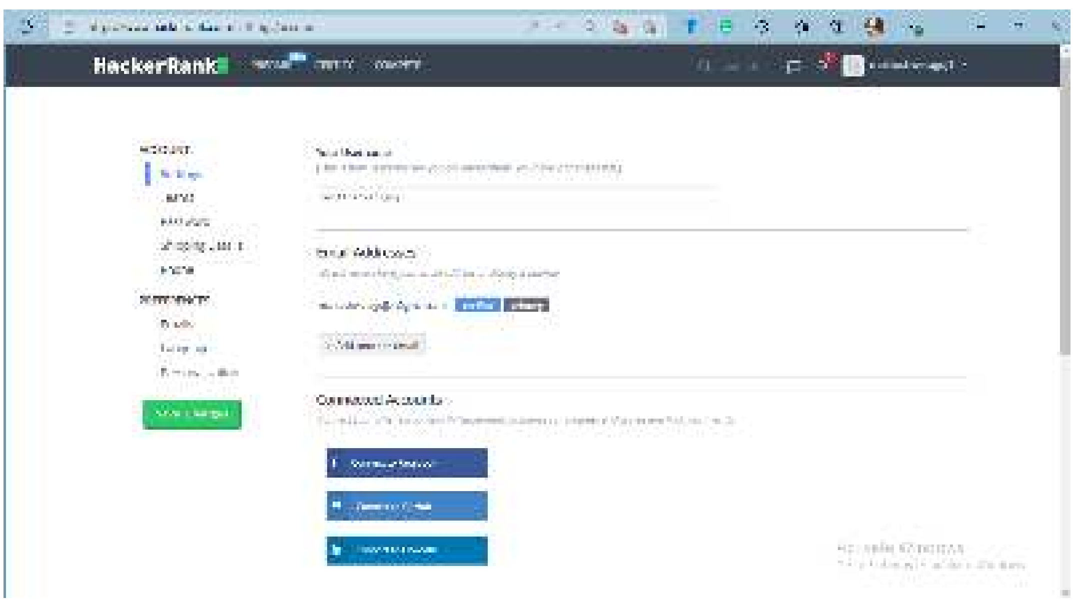


Figure 13. Hackerrank Settings Page

Congratulations! Your Coding Journey has started.

Classwork Problems

Week 1

URL: Week 1

Problem 1: M1/P1/L1/8/GCD

Problem Statement: Consider a scenario for 'T' test cases. For each test case 't1', let 'a1' and 'a2' be two positive number and there exist a scenario where 'a1' and 'a2' are divisible by a set of positive numbers S={x1, x2,...xn}.
Compute the greatest number 'x' from the set 'S', such that 'x' divides both 'a1' and 'a2'.

Input Format
The first line contains an integer T, total number of test cases.
Then follow T lines, each line contains an integer a1 and a2.

Constraints
1<=T<=1000
1<=a1,a2<=100000

Output Format
The value of x.
Do not print additional messages.

Sample Input 0
3
5 15
4 24
3 7

Sample Output 0
5
4
1

Explanation 0
Number of test cases = 3
For test case t1, value of a1 = 5, value of a2 = 15
Greatest Common Divisor of a1 and a2 = 5
Similarly for test case t2 and t3.

Problem 2: M1/P1/L1/9/Sum of Factorial

Problem Statement: Consider a scenario with 'T' test cases. Hence, for each test case 'ti', let there be any positive number 'n'. Since, factorial of a number 'n' can be computed through the following formula:
$n! = n \times (n - 1) \times ... \times 3 \times 2 \times 1$

Find a value, named as 'result', by adding all the digits received through a factorial of a number 'n'.

Input Format
The First line contains T, the number of test cases, followed by integer 'n'.

Constraints
1<=T<=1000
1<=n<=1000000

Output Format
Sum of the digits of a computed factorial value.

Sample Input 0
2
5
8

Sample Output 0
3
9

Explanation 0
Number of test cases = 2
For test case t1, value of 'n' = 5
n! = 5*4*3*2*1 = 120
result = 1+2+0 = 3
Similarly, for test cases t2.

Problem 3: M1/P1/L1/Perfect number

Problem Statement: Let, 'n' be any positive number, such that it has a set of positive divisors 'S' = {x1, x2,..., xn}. Hence, a number 'n' is said to be as perfect number, if and only if the sum of all the elements forms the number 'n' itself.

Input Format
A number 'n'.

Constraints
1≤n≤10000

Output Format
Use 'yes', if condition is satisfied, 'no' otherwise.
Do not include additional messages.

Sample Input 0
6

Sample Output 0
yes

Explanation 0

Let 'n' = 6
Factors of 'n', set S = {1, 2, 3}
Sum = 1+2+3 = 6
Hence, 6 is a perfect number.

Week 2

URL: Week 2

Problem 1: M1/P1/L1/5/Sum of Prime

Problem Statement: Consider a scenario with 'T' test cases, where, for each test case 'ti' there exists a number 'n', such that it holds a positive integer value. Hence, a set 'S' is formed out from the series that holds a constraint that:
S=(x1, x2, x3,...)
Every element in the set 'S' is prime in nature.
Hence, 'result' can be concluded as (x1+x2+x3+...)

Input Format
The First line contains T, the number of test cases, followed by integer N.
Do not print additional messages.

Constraints
1<=T<=1000
1<=N<=1000000

Output Format
The value of 'result' for each number 'n'.
Do not print additional messages.

Sample Input 0
3
10
30
50

Sample Output 0
17
129
328

Explanation 0
Number of test cases, T=3
for test case 't1', n = 10
Set S = (2, 3, 5, 7)
Value of result = 2+3+5+7 = 17
Similarly, for test case t2 and t3.

Problem 2: M1/P1/L1/11/Palindrome

Problem Statement: Consider a scenario, where there exists two positive numbers 'n1' and 'n2'. The numbers 'n1' and 'n2' follows certain set of constraints:
They are always positive
They can never hold a value greater than 100

Compute the result 'r' using the formula given below:
r = max   (n1*n2) for all 10<=n1,n2<100, such that 'r' is equivalent to its reverse.

Input Format
The maximum value of 'n1' and 'n2'.
Do not print additional messages.

Constraints
10 <= n1, n2 < 100
Both are always positive.

Output Format
The result 'r', if it validates the state, print "Not a valid state" in case of non-valid state.
Do not print additional messages.

Sample Input 0
99
99

Sample Output 0
9009

Explanation 0
Value of 'n1' = 99
Value of 'n2' = 99
Hence, result 'r' = max (n1 * n2) and 'r' is a palindrome.

Value of 'n1'	Value of 'n2'	Result 'r'
10	10	100
10	11	110
10	12	120
.	.	.
.	.	.
.	.	.
99	99	9801

Problem 3: M1/P1/L1/1/Play with Numbers

Problem Statement: Write a program to obtain a number N, Such that: Case1: increment it's value by 2 if it is divisible by 11 Case2: decrement it's value by 2 if it is divisible by 4 Case3: else multiply the number by 4

Input Format
The First line contains T, the number of test cases, followed by integer N.

Constraints
1<=T<=1000 1<=N<=100000

Output Format
The new number in new line.

Sample Input 0
3
121
64
15

Sample Output 0
123
62
60

Explanation 0
N=121, it is divisible by 11 so it will fall in case 1 hence updated number is 121+2=123

Week 3

URL: Week 3

Problem 1: M1/P1/L1/26/complement of number

Problem Statement: Suppose, you are given a number 'num', then, there exists a function 'find()' such that it takes an input of type 'integer' and flips its binary representation to compute the 1's complement of a number.

Input Format
num ranges (1,100) There should be no messages (such as, 'Enter the number:').

Constraints
Identifiers for each of the fucntion and variables are case-sensitive, i.e., 'T' and 't' are different.

Output Format
Output should not contain any message (such as, 'Output is:'). It should strictly follow the format provided.

Sample Input 0
5

Sample Output 0
2

Explanation 0
Here, in this example,
Sample Input = 5,
its Binary Representation is 101
1's Complement is 010
Sample Output = 2

Problem 2: M1/P1/L1/18/count of divisibility

Problem Statement: You have been given 3 integers - l, r and k. Find how many numbers between l and r (both inclusive) are divisible by k. You do not need to print these numbers, you just have to find their count.

Input Format
Integers for l,r,k.

Constraints
1<=l,r,k<=10^5

Output Format
Output should be in the form of integer. In some case if the value of k >l,r print a message "value of k is supposedly larger"

Sample Input 0
4
9
2

Sample Output 0
3

Explanation 0
L=4,r=9,k=2 there are 3 numbers between 4 and 9 that are divisible by 2

Problem 3: M1/P1/L1/16/Next_term

Problem Statement: Given a number N, find the Nth term in the series 1, 3, 6, 10, 15, 21…

Input Format
The Number n

Constraints
1

Output Format
The nth digit of series.

Sample Input 0
10

Sample Output 0
55

Week 4

URL: Week 4

Problem 1: M1/P5/L1/3/Sum of Digits until Single Digit

Problem Statement: We define Digital_Sum of an integer using the following rules:
Given an integer, we need to find the Digital_Sum of the integer.
If x has only 1 digit, then its Digital_Sum is x. Otherwise, the Digital_Sum of x is equal to the Digital_Sum of the sum of the digits of x. For example, the Digital_Sum of 9875 will be calculated as:
Digital_Sum(9875)   9+8+7+5 = 29
Digital_Sum(29)   2 + 9 = 11
Digital_Sum(11)   1 + 1 = 2
Digital_Sum(2)   = 2

Input Format
given a number defined by n.

Constraints
1<=n<=100000

Output Format
Sum of digit until unit digit.

Sample Input 0
123

Sample Output 0
6

Problem 2: M1/P6/L2/6/ find the sum of digit in a string.

Problem Statement: You have to enter a string of length L which may consists some digit as well. You need to find that digits and print the sum of all digits present in the string.

Input Format
You have to enter a string of length L
for example:
50
hello1 world23 its a pr5ogr4m6

Constraints
0

Output Format
Output should display the digits present in the string first and then the sum of digits in next line.

Sample Input 0
30

hello1 world23 its a pr5ogr4m6

Sample Output 0
The digits present in the string is 123546
The sum of digits is 21

Problem 3: Armstrong Numbers 11

Problem Statement: An Armstrong number is a number that is sum of its own digits each raised to the power of the number of digits. Write a program to print all the Armstrong numbers in given range

Input Format
First line consists of an integer T which is the number of testcases.And then there will be T line-separated numbers each line consisting of upper bound and lower bound to print all the Armstrong numbers.

Constraints
1

Output Format
T number of line separted armstrong number,each line consisting of armstrong number(s)

Sample Input 0
1
100 999

Sample Output 0
153 370 371 407

Week 5

URL: Week 5

Problem 1: M1/P2/L2/33/Write a program in C to delete an element at desired position from an array

Problem Statement: You will be given an array of size 6 and we have to delete an element from second position.

Input Format
1 2 3 4 5 6

Constraints
N=6

Output Format
1 2 4 5 6

Sample Input 0
1 2 3 4 5 6

Sample Output 0
1 2 4 5 6

Problem 2: M1/P2/L1/6/print all unique elements in an array

Problem Statement: You will be given an array for which you have to print all unique elements in that array

Input Format
10 numbers separated by blank space

Constraints
Total elements N=10
Each element can vary between 1 to 10 (both inclusive)

Output Format
Unique numbers separated by blank space

Sample Input 0
3 4 5 5 6 7 7 6 7 6

Sample Output 0
3 4

Problem 3: M1/P2/L1/8/count the frequency of each element of an array

Problem Statement: In this program, we need to count the occurrence of each unique element present in the array. One of the approach to resolve this problem is to maintain one array to store the counts of each element of the array. Loop through the array and count the occurrence of each element and store it in another array.
1 2 4 7 2 2 2 4 1
In the above array, 1 has appeared 2 times, so, the frequency of 1 is 2. Similarly, 2 has appeared 4 times. The frequency of 2 is 4 and so on.

Input Format
1 2 4 7 2 2 2 4 1

Constraints
Total number of elements N=9
Each element lie between 1 to 99

Output Format
1 2
2 4
4 2
7 1

Sample Input 0
1 2 4 7 2 2 2 4 1

Sample Output 0
1 2
2 4
4 2
7 1

Explanation 0
Frequency of each number in the input

Problem 4: M1/P2/L1/16/find the largest and second largest element in an array.

Problem Statement: You will be given an array of size 6 for which you have to find the largest and second largest element in an array.

Input Format
1 2 5 6 3 2 2 3 4 6 7 8

Constraints
N=6

Output Format
6 5 8 7

Sample Input 0
1 2 5 6 3 2

Sample Output 0

6
5

Problem 5: Crossword Puzzle 1

Problem Statement: You need to play a crossword puzzle.

You are given a 2D square matrix of characters. Your task is to find whether a given word can be found in the matrix by going from left to right or from top to bottom.
Example,
[['F', 'A', 'C', 'E'],
 ['A', 'B', 'C', 'D'],
 ['C', 'A', 'O', 'B'],
 ['T', 'N', 'S', 'O']]
If the required word is "FACE", then it can be found in row 0.
If the required word is "BAN", then it can be found in col 1 (starting from row 1).
If the required word is "FACT", then it can be found in col 0.

Input Format
The first line contains size of the square matrix N
The next N lines contains element of the 2D square matrix.
The last line contains the target word that need to be searched.

Constraints
The matrix is square

Output Format
The output is either True or False depending on whether the word is present in matrix or not.

Sample Input 0
4
FACE
ABCD
CAOB
TNSO
BAN

Sample Output 0
True

Sample Input 1
4
FACE
ABCD
CAOB
TNSO
FACE

Sample Output 1
True

Week 6

URL: Week 6

Problem 1: M1/P2/L2/22/find transpose of a given matrix

Problem Statement: Transpose of a matrix is obtained by changing rows to columns and columns to rows. In other words, transpose of A[][] is obtained by changing A[i][j] to A[j][i]. Consider the size of matrix N*N. Where N is 3 (fixed)

Input Format
1 2 3 4 5 6 7 8 9 // 1 2 3 elements of first row, 4 5 6 elements of second row, 7 8 9 elements of third row
1 4 0 -5 2 7 //1 4 0 elements of first row, -5 2 7 elements of second row

Constraints
0<N<100

Output Format
1 4 7 2 5 8 3 6 9 // 1 4 7 elements of first row, 2 5 8 elements of second row, 3 6 9 elements of third row 1 -5 4 2 0 7 //1 -5 elements of first row, 4 2elements of second row, 0 7 elements of third row

Sample Input 0
1 2 3 4 5 6 7 8 9

Sample Output 0
1 4 7 2 5 8 3 6 9

Problem 2: M1/P2/L1/3/sum of all elements of the array

Problem Statement: Suppose you have A attempts and in each attempt you will be given an array for which you have to find the sum of all elements of array.

Input Format
Enter the size of array: 5 1 5 4 3 2

Constraints
0<N<100

Output Format
15

Sample Input 0
1
2
3
3
2

Sample Output 0

11

Problem 3: M1/P2/L1/22/Write a program in C to rotate an array by N positions

Problem Statement: You are given an array A of size size, Wherein you are supposed to code a program that allows the user to shift the array by n positions either to left or to right.

Input Format

The first line shall include the aray size , that will be in positive integer. Second line shall include the number of rotations. The third line and follwing them should have the array elements.

Constraints
1<size<=10

Output Format
The output should be in two lines 1st: the original array 2nd: the shifted array

Sample Input 0
5
2
10
20
30
40
50

Sample Output 0
10 20 30 40 50
40 50 10 20 30

Explanation 0
5 being the size of the array, 2 being the number of rotations To the RIGHT (you are required to do the same). Mind the space according to the sample output.

Problem 4: M1/P2/L1/24/Write a program in C to find the two repeating elements in a given array.

Problem Statement: You are given an array A[] of maximum size 10, you need to write a program that will enable you to find the duplication of elements in that array.

Input Format
Integer for size.

Constraints
1<=size<=10
Each array element will vary between 1 to 10 (both inclusive)

Output Format
Output will be both the numbers of duplication.

Sample Input 0

5
1
2
3
4
4

Sample Output 0
4 4

Explanation 0
For given size 5 , elements are entered. 4 occurs in duplication in that array. Hence 4 4

Problem 5: Overlapping Series

Problem Statement: You have N series. You are given starting point ai and ending point bi for each series.
You need to identify and remove the redundant series i.e., the series which is part of a larger series.

Input Format
The first line contains N : The number of series
The next N lines contains the starting point ai and ending point bi of each series

Constraints
2 <= N <= 100
1 <= ai , bi <= 10000

Output Format
The starting and ending point of non-overlapping series in the order they were entered

Sample Input 0
4
1 3
5 8
4 10
20 25

Sample Output 0
1 3
4 10
20 25

Explanation 0
The series (5, 8) is entirely overlapped by a larger series (4, 10) so it is removed

Problem 6: Plus One 6

Problem Statement: You are given a large integer represented as an integer array digits, where each digits[i] is the ith digit of the integer. The digits are ordered from most significant to least significant in left-to-right order. The large integer does not contain any leading 0's.
Increment the large integer by one and return the resulting array of digits.

Input Format
an integer number N

Constraints
$0<N<10000$
$0<M<10$
where M is the number of digits in a number.

Output Format
incremented array

Sample Input 0
123

Sample Output 0
124

Explanation 0
The array represents the integer 123. Incrementing by one gives 123 + 1 = 124

Week 7

URL: Week 7

Problem 1: M1/P4/L1/1/Sum of First and Last Digits of a Number

Problem Statement: Suppose you have T attempts and in each attempt, you will be given numbers for which you have to find the sum of first and last digit of a number.

Input Format
T attempts for each attempt a number will be given and defined by N.
for example:
3
5656
65657676766787676
65745356789

Constraints
0<T<100
0<N<10^18

Output Format
sum of each attempt in next line.

Sample Input 0
3
2365
6547
6987

Sample Output 0
7
13
13

Explanation 0
for example: in 1st attempt the number is 2365, and first and last digit summation is 2+5=7.

Problem 2: M1/P4/L1/3/Sum odd digits

Problem Statement: Suppose you have T attempts and in each attempt, you will be given a number for which you have to find the sum of odd positional digits of that number.

Input Format
T attempts, for each attempt a number will be given and defined by N. for example:
3
5656
65657676766787676
65745356789

Constraints
0<T<100
0<N<10^18

Output Format
sum of each attempt in next line.

Sample Input 0
3
2365
65471
6987

Sample Output 0
8
11
14

Problem 3: M1/P4/L1/4/sum of modulo K of first N natural number

Problem Statement: Suppose you have T attempts and in each attempt, you will be given two numbers K and N to find the sum of modulo K of first N natural numbers.

Input Format
T attempts each attempt have a two numbers K and N for which have to find the sum of modulo K of first N natural numbers.
for example:
3
40 245
12 356
34 4525464

Constraints
0<T<100
0<K<100
0<N<10^9

Output Format
each attempt output space separated on next line.

Sample Input 0
2
40 245
12 356

Sample Output 0
4695
1950

Explanation 0

Input : N = 10 and K = 2.
Output : 5
Sum = 1%2 + 2%2 + 3%2 + 4%2 + 5%2 + 6%2 +
    7%2 + 8%2 + 9%2 + 10%2
  = 1 + 0 + 1 + 0 + 1 + 0 + 1 + 0 + 1 + 0
  = 5
similarly; if K 12 and N=356 then answer is 1950.

Problem 4: M1/P4/L1/5/Count the Fibs?

Problem Statement: Let us define a Fibonnaci numbers as:
f1 = 1
f2 = 2
fn = fn-1 + fn-2
You are given two numbers x and y, find the Fibonnaci numbers that lie in the range (x,y).
Note the first Fibonnaci number is assumed to be 1

Input Format
There are T attempts. For each attempt you are given two non-negative inputs x and y. For example:
3
10 100
5 20
1 1000

Constraints
x <= y <= 1010

Output Format
For each attempt, the output is displayed on a single line.

Sample Input 0
2
10 100
1234567890 9876543210

Sample Output 0
5
4

Problem 5: M1/P4/L1/2/Carmichael Numbers

Problem Statement: A number n is said to be a Carmichael number if it satisfies the following modular arithmetic condition:
power(b, n-1) MOD n = 1,
for all b ranging from 1 to n such that b and
n are relatively prime, i.e, gcd(b, n) = 1
Suppose you have T attempts and in each attempt, you will be given N numbers to find the Carmicheal number of each.

Input Format

T attempts each attempt have a number N for which we have to find whether it is Carmichael number or not.
for example:
3
561
8
2365

Constraints
0<T<100
0<N<10^9

Output Format
each attempt output space separated on next line.

Sample Input 0
3
561
8
2365

Sample Output 0
true
false
false

Explanation 0
Here Attempts are 3,
Input : n = 8 Output : false Explanation : 8 is not a Carmichael number because 3 is relatively prime to 8 and (3^8-1) % 8 = 2187 % 8 is not 1.
Input : n = 561 Output : true

Problem 6: M1/P4/L1/7/ calculate nCr

Problem Statement: Suppose you have T attempts and in each attempt, you will be given two numbers for which you have to find the nCr.
Logic: nCr = (n!) / (r! * (n-r)!)

Input Format
T attempts for each attempt two numbers space separated will be given. for example: 3 5 2 3 1

Constraints
0<T<100
0<n<1000
0<r<1000

Output Format
each attempt nCr output on next line.

Sample Input 0
2

5 2
3 1

Sample Output 0
10
3

Explanation 0
here Attempts are T=3 for first attempt, n=5 and r=2
then (5C2)=10

Week 8

URL: Week 8

Problem 1: M1/P4/L1/8/ Square root of a number by Repeated Subtraction method

Problem Statement: Suppose you have T attempts and in each attempt, you will be given a number for which you have to find the Square root of a number by Repeated Subtraction method.
Logic:
N = 81
Step 1: 81-1=80
Step 2: 80-3=77
Step 3: 77-5=72
Step 4: 72-7=65
Step 5: 65-9=56
Step 6: 56-11=45
Step 7: 45-13=32
Step 8: 32-15=17
Step 9: 17-17=0
Since, 9 odd numbers were used, hence the square root of 81 is 9.
If n is not a perfect square then the output will be floor integer.

Input Format
T attempts for each attempt a number n will be given. for example:
3
81
72
65

Constraints
0<T<100
0<n<1000

Output Format
each attempt output on next line.

Sample Input 0
3
81
72
65

Sample Output 0
9
8
8

Explanation 0
Here T is 3, and first n value is 81.

for which :
N = 81
Step 1: 81-1=80
Step 2: 80-3=77
Step 3: 77-5=72
Step 4: 72-7=65
Step 5: 65-9=56
Step 6: 56-11=45
Step 7: 45-13=32
Step 8: 32-15=17
Step 9: 17-17=0
Since, 9 odd numbers were used, hence the square root of 81 is 9.

Problem 2: M1/P4/L1/9/ Multiplication of a number with its complement.

Problem Statement: Suppose you have T attempts and in each attempt, you will be given a number for which you have to find the multiplication with its complement.

Input Format
T attempts for each attempt a number n will be given.
for example:
3
17
68
25

Constraints
0<T<10
0<n<100

Output Format
each attempt output on next line.

Sample Input 0
3
17
68
25

Sample Output 0
-306
-4692
-650

Explanation 0
Here T=3, and first n number is 17. and the complement of n is -18, so that result is 17*-18= -306

Problem 3: M1/P4/L1/11 GCD IN ARRAY

Problem Statement: The candidate is required to take input x from user in an array of size N. Once the inputs x are taken and displayed, the gcd is to be calculated of the two largest elements of the array.

Input Format
First line should have the number of inputs N. Second line onwards there are N inputs denoting various elements of the array

Constraints
2<=N<=10^5
1<=x<=10^5
Negative Values as of for elements are acceptable. Float values as of for elements of array are not acceptable.

Output Format
only the GCD

Sample Input 0
5
2
4
5
10
20

Sample Output 0
10

Explanation 0
INPUT:- 5 (number of elements) rest are the inputs in the arary. OUTPUT:- The GCD of those two largest elements.

Problem 4: M1/P4/L1/6/Count the sandwiches

Problem Statement: Tony Stark has n sandwiches. He eats them one by one by removing the crust of the sandwich. From k>1 sandwich crusts, he can make a new sandwich.
How many sandwiches can Tony Stark have?

Input Format
There are T attempts. In each attempt the input consists of two integer numbers representing n and k.

Constraints
0 < T < 100
0 < N < 10 10
0 < k < 1000

Output Format
Each attempt output is displayed on next line.

Sample Input 0
4
10 2
100 3
1000 4

10000 5

Sample Output 0
19
149
1333
12499

Problem 5: M1/P4/L1/25/Multiplication of First max and Second max

Problem Statement: You are required to take N number of inputs so as to multiply binary numbers.Once you have all the binary numbers, you just need to calculate multiplication of 1st max and 2nd max binary number. And print the output in binary again.

Input Format
3<=N<=20(N is the number of inputs)

Constraints
Binary number <=10^5

Output Format
Binary Number

Sample Input 0
3
10
1
11

Sample Output 0
110

Explanation 0
The binary input is 10,1,11 and the multiplication of 2 and 3 yields 6 i.e, 110

Week 9

URL: Week 9

Problem 1: M1/P6/L1/10 (C program to count the number of characters in a given string.)

Problem Statement: Consider a code, where there exist a function str_c(), that takes an input of type 'string' and returns the total count of characters in that particular stirng.

Input Format
Input string should not have characters more than 500.

Constraints
Identifiers for each of the fucntion and variables are case-sensitive, i.e., 'T' and 't' are different.

Output Format
Output should strictly follow the format provided as Sample Output.

Sample Input 0
'Hello World'

Sample Output 0
13

Explanation 0
Here, in the example provided,
Sample Input = 'Hello World'
Sample Output = 13

Problem 2: M1/P6/L1/16 (C program to reverse a given string.)

Problem Statement: Consider a code, where there exist a function str_r() which takes an input type 'string' of length 'n' and in return finds a string that occurs in revrse order.

Input Format
Input string should not have characters more than 500. There should be no messages (such as, 'Enter the string:').

Constraints
Identifiers for each of the fucntion and variables are case-sensitive, i.e., 'T' and 't' are different.

Output Format
Output should not contain any message (such as, 'Output string is:'). It should strictly follow the format provided.

Sample Input 0
'Hello World'

Sample Output 0

'dlroW olleH'

Explanation 0
Here, in this example,
Sample Input = 'Hello World'
Sample Output = 'dlroW olleH'

Problem 3: M1/P6/L1/26 (C program to limit the count of characters entered by user.)

Problem Statement: Suppose, you are given a string 'str' of length 'n' and you are allowed to input the count of characters 'm', where 'm' is never greater than 'n' and n=500.
Consider a function named as str_c(), that takes a number 'm' as input, and produces the equivalent set of characters of length 'm'.

Input Format
There should be no messages (such as, 'Enter the string:').

Constraints
Identifiers for each of the fucntion and variables are case-sensitive, i.e., 'T' and 't' are different.

Output Format
Output should not contain any message (such as, 'Output string is:').
It should strictly follow the format provided.

Sample Input 0
3
Hey

Sample Output 0
Hey

Explanation 0
Here, in this example,
m=3
Sample Input String = hey
Sample Output String = hey

Problem 4: M1/P6/L1/34 (C program to print the ASCII values of all the character in a given string.)

Problem Statement: Consider a string 'str' of length 'n', such that it can include digits, alphabets and special characters (or combination of these three). There exists a function str_c() that takes some input of type string and prints the equivalent decimal values for every character representation that lies in a particular given string.

Input Format
Input string should not have characters more than 500. There should be no messages (such as, 'Enter the string:').

Constraints
Identifiers for each of the function and variables are case-sensitive, i.e., 'T' and 't' are different.

Output Format
Output should not contain any message (such as, 'Output string is:'). Outputs should be separated by a white space. It should strictly follow the format provided.

Sample Input 0
Hello World

Sample Output 0
72 101 108 108 111 32 87 111 114 108 100

Explanation 0
Here, in this example,
Sample Input String = Hello World
Sample Output = 72 101 108 108 111 32 87 111 114 108 100

Problem 5: M1/P6/L1/8/ count number of words

Problem Statement: You have to enter a sentence of length L, and count the number of words It has

Input Format
You have to enter a string of length L for example: 20 I am a student

Constraints
0

Output Format
Output should display in next line

Sample Input 0
14
I am a student

Sample Output 0
The number of words = 4

Problem 6: M1/P6/L2/6/ find the sum of digit in a string.

Problem Statement: You have to enter a string of length L which may consists some digit as well. You need to find that digits and print the sum of all digits present in the string.

Input Format
You have to enter a string of length L
for example:
50
hello1 world23 its a pr5ogr4m6

Constraints
0

Output Format

Output should display the digits present in the string first and then the sum of digits in next line.

Sample Input 0
30
hello1 world23 its a pr5ogr4m6

Sample Output 0
The digits present in the string is 123546
The sum of digits is 21

Week 10

URL: Week 10

Problem 1: M1/P3/L1/1.Toppers of class

Problem Statement: There is a class of students, these students have recently appeared for the final exams. Top 10 students of the class have been selected for being Class Representatives in the next semester. Provide the list of those 10 top marks of the same class. Also identify the top 3 of them which are the Class Representatives.

Input Format
Input for the number of students.
Int Array to store the marks.

Constraints
1<Array[size]<=30
Array[element]<=100

Output Format
The first row represents top 10 scores seperated by comma.
The next row represnt top 3 scores of Class Represntatives seperated by comma.

Sample Input 0
12
76
87
95
81
65
89
99
45
90
49
77
34

Sample Output 0
99,95,90,89,87,81,77,76,65,49,
99,95,90,

Explanation 0
The list in first row shows the top 10 students
The list in second row shows the top 3 students selected as class representatives.

Problem 2: M1/P3/L1/2.DANCE PERFORMANCE

Problem Statement: Consider a school annual fest that has a series of events ,one of the event is dance performance. The number of students on the stage depends on length and width of the stage.The number of maximum students that can accommodate on the stage is 20. The maximum rows that the stage has is 4. That is each row can have 5 students. We have to arrange these students such that all of them are able to view the dance performance properly. Arrange them in increasing order of height in a way that the shortest one are in the first row, taller than the ones that were in the first row are in second row and so on, then the tallest students are in the last row.

Input Format
float Array to store heights of students

Constraints
2 <= Array[element] <= 7

Output Format
4 separate rows , 1st row having shortest height and so on and 4th row at then end has tallest students

Sample Input 0
3.6
5.9
7
6.7
5.9
2.7
3.6
5.8
2.6
3.7
4.9
4.11
5.11
6.9
5.8
6
5.8
3.6
2.8
4.8

Sample Output 0
2.60,2.70,2.80,3.60,3.60,
3.60,3.70,4.11,4.80,4.90,
5.11,5.80,5.80,5.80,5.90,
5.90,6.00,6.70,6.90,7.00,

Explanation 0
Total number of students is 20, with number of rows =4 . Each row should havce exactly 5 heights in the increasing order.

Solution:

Problem 3: M1/P3/L1/3.CAB SERVICE

Problem Statement: One of the IT companies in the city has many employees. For the convenience of the employees the company arranges the cab and the bus facility accordingly. Availing the cab facility depends on the vicinity . On every round of the cab only 10 employees can avail the cab facility in such a way that the one staying nearest is the first to be picked. We are required to provide the list of the distances to the drives in order to help him out get the best path.

Input Format
Array of integers

Constraints
Array[size]<=10 5

Output Format
closest to most far away distance

Sample Input 0
5
30
44
50
23
11

Sample Output 0
11,23,30,44,50,

Explanation 0
This example is done by considering the array of size 5, The array is starting from closest distance to most faraway distance

Problem 4: M1/P3/L1/6.Investing money

Problem Statement: Richard wants to invest some money , for this Richard enters some inputs. After which Richard enters the actual amount x. Now we want that if x^3 is present in the previous entered then return the x^3. Make use of the amt_return().

Input Format
Take input x as an integer Array of integer values.

Constraints
1

Output Format
Either x or x^3 according to the input

Sample Input 0
4
65
76
87
98
64

Sample Output 0
64

Explanation 0

The first line has the amount. The next 5 lines have the array elements(in sample we have taken array of 5 , for rest test cases kindly take array size of 10). The output has returned 64 as 4^3 was encountered in the array.

Problem 5: M1/P3/L1/4.Cycling competition

Problem Statement: There is an inter university cycling competition , there are many participants who are trying very hard to win the prize that the university has decided. Every college has to provide the highest speed of the cyclists of their respective colleges to the university. There are x participants in Kiet who have to participate , help the college to get the speed of the fastest cyclist of the college.

Input Format

Number of participants Integer type Array to store speeds

Constraints

Array[size]<=30 Array[element]<=30

Output Format

Maximum speed

Sample Input 0

5
20
24
28
10
7

Sample Output 0

28

Explanation 0

The first line has the number of cyclists. Following line shave the speeds Last line has the Maximum speed

Week 11

URL: Week 11

Problem 1: M1/P5/L1/1/Factorial(Using Recursion)

Problem Statement: The factorial function (symbol: !) says to multiply all whole numbers from our chosen number down to 1.
Examples: 4! = 4 × 3 × 2 × 1 = 24 7! = 7 × 6 × 5 × 4 × 3 × 2 × 1 = 5040 1! = 1
Remember factorial of Zero(0!) = 1. What About Negatives? Can we have factorials for numbers like −1, −2, etc? No. Negative integer factorials are undefined.

Input Format
T Attempts for each attempt, you will be given a number defined by N For Example: 3 //Number of Attempts 0 //Factorial of 0 is 1 5 // Factorial of 5 is 120 6 // Factorial of 6 is 720

Constraints
1

Output Format
Factorial of 0 is 1 Factorial of 5 is 120 Factorial of 6 is 720

Sample Input 0
3
5
0
6

Sample Output 0
Factorial of 5 is 120
Factorial of 0 is 1
Factorial of 6 is 720

Explanation 0
According to problem statement
T Attempts for each attempt, you will be given a number defined by N For Example: 3 //Number of Attempts 0 //Factorial of 0 is 1 5 // Factorial of 5 is 120 6 // Factorial of 6 is 720

Problem 2: M1/P5/L1/10/Print sum of First N natural numbers using recursion.

Problem Statement: Print sum of First N natural numbers using recursion. in N <1 print "Invalid natural number"

Input Format
No. of attempt N //natural number

Constraints
1<=T<=100 1<=N<=100

Output Format
Print sum of First N natural numbers for each attempts in new lines

Sample Input 0
5

Sample Output 0
15

Problem 3: M1/P5/L1/11/Reverse a word using Recursion

Problem Statement: Reverse a word using Recursion

Input Format
Number of attempts word(An array of characters)

Constraints
1<=T<=100 word(An array of characters)

Output Format
Print reveresed word in new lines for aech attempts

Sample Input 0
2
hello
hi

Sample Output 0
olleh
ih

Sample Input 1
2
computer
science

Sample Output 1
retupmoc
ecneics

Problem 4: M1/P5/L1/2/Multiply two numbers using Recursion

Problem Statement: Multiply two number using Recursion Examples:
2*5 = 10
-2*5 =-10
1*0= 0

Input Format
T Attempts for each attempt, you will be given a number defined by N
For Example:
3 //Number of Attempts

2 5 //Two numbers for first attemp
-2 5 // Two numbers for second attemp
1 0 // Two numbers for third attemp

Constraints
N1 and N2 are two number to be multiply. T is number of attempts
0<T<10
-10<N1<100
-10<N2<100

Output Format
10
-10
0

Sample Input 0
1
2 3

Sample Output 0
6

Problem 5: M1/P5/L1/3/Sum of Digits until Single Digit

Problem Statement: We define Digital_Sum of an integer using the following rules:
Given an integer, we need to find the Digital_Sum of the integer.
If x has only 1 digit, then its Digital_Sum is x. Otherwise, the Digital_Sum of x is equal to the Digital_Sum of the sum of the digits of x. For example, the Digital_Sum of 9875 will be calculated as:
Digital_Sum(9875) 9+8+7+5 = 29
Digital_Sum(29) 2 + 9 = 11
Digital_Sum(11) 1 + 1 = 2
Digital_Sum(2) = 2

Input Format
given a number defined by n.

Constraints
1<=n<=100000

Output Format
Sum of digit until unit digit.

Sample Input 0
123

Sample Output 0
6

Home Assignment Problems

Week 1

URL: [Week 1]

Problem 1: Frequency Count

Problem Statement: You are given a string and your task is to print the sum of upper case characters and lower case characters individually.

Input Format
String consisting of only upper case and lower case characters.

Constraints
1<=length of string<=1000

Output Format
Sum of upper characters and sum of lower characters separated by a space.

Sample Input 0
aAcbdDF

Sample Output 0
3 4

Explanation 0
upper case character in this test case is A,D,F. so count is 3.
lower case character in this test case is a,c,b,d. so count is 4.

Problem 2: M1/P4/L1/16/A Power B Power C

Problem Statement: For T attempts by the user the code should find the output of For T attempts by the user the code should find the output of a^{b^c} modulo 10^9 +7.

Input Format
The first input line has an integer n: the number of calculations. After this, there are n lines, each containing three integers a, b and c.

Constraints
1≤n≤10^5 0≤a,b,c≤10^9

Output Format
Print each value a^{b^c} modulo 10^9 + 7.

Sample Input 0
3
3
7
1
15
2
2
3
4
2

Sample Output 0
2187
50625
43046721

Explanation 0
Input: 3 attempts a=3, b=7, c=1 a=15, b=2, c=2 a=3, b=4, c=2 Output: 2187 50625 43046721

Problem 3: M1/P4/L1/18/Mango Distribution

Problem Statement: There are n children and m mangoes that will be distributed to them. Your task is to count the number of ways this can be done. For example, if n=3 and m=2, there are 6 ways: [0,0,2], [0,1,1], [0,2,0], [1,0,1], [1,1,0] and [2,0,0].

Input Format
The only input line has two integers n and m.

Constraints
$1 \leq n,m \leq 10^6$

Output Format
Print the number of ways modulo 10^9 +7.

Sample Input 0
3
2

Sample Output 0
6

Explanation 0
For example, if n=3 and m=2, there are 6 ways: [0,0,2], [0,1,1], [0,2,0], [1,0,1], [1,1,0] and [2,0,0].

Week 2

URL: Week 2

Problem 1: M1/P4/L1/19/Gift Distribution

Problem Statement: There are n children at a Christmas party, and each of them has brought a gift. The idea is that everybody will get a gift brought by someone else. In how many ways can the gifts be distributed?

Input Format
The only input line has an integer n: the number of children.

Constraints
1<=n<=10^6

Output Format
Print the number of ways modulo 10^9 +7

Sample Input 0
4

Sample Output 0
9

Explanation 0
For input As 4 students, there are 9 ways in which they can exchange the gifts such that none receives their own gift.

Sample Input 1
10

Sample Output 1
1334961

Problem 2: M1/P1/L1/21/numbercircle

Problem Statement: You need to code for the pattern that form the circle of numbers

Input Format
The integer for which the pattern needs to be formed.

Constraints
1<=n<=10

Output Format
pattern as in the sample

Sample Input 0

4

Sample Output 0
4 4 4 4 4 4 4
4 3 3 3 3 3 4
4 3 2 2 2 3 4
4 3 2 1 2 3 4
4 3 2 2 2 3 4
4 3 3 3 3 3 4
4 4 4 4 4 4 4

Explanation 0
Count the number of rows and columns carefully accordingly.

Problem 3: M1/P4/L1/22/Prime Factors

Problem Statement: Write a program that, given a positive integer N, returns the number of its prime factors. For example, given N = 24, the function should return 2, because 24 has 2 prime factors, namely 2, and 3. There are no other factors of 24.

Input Format
Input an integer N.

Constraints
N is an integer within the range [1..2,147,483,647].

Output Format
Output will be in integer format.

Sample Input 0
10

Sample Output 0
2

Explanation 0
For the input of 10 there are 2 prime number that are 2 and 5.

Week 3

URL: Week 3

Problem 1: Altered Fibonacci

Problem Statement: Rishabh is novice in programming. He learned how to generate a Fibonacci series. He noticed that Fibonacci series is always increasing so he altered it such that f(n) = | f(n-1) - f(n-2) |. It means that f(n) can never be negative. So all he wants is to count the number of iterations that series take in reducing itself to zero.

Input Format
The first line of the input contains an integer T denoting the number of test cases. The description of T test cases follows. Only one line of each test case, contains two integers f(0) and f(1) denoting the number described above.

Constraints
1 ≤ T ≤ 10
1 ≤ f(0) ≤ 10^4
1 ≤ f(1) ≤ 10^4

Output Format
For each test case, output a single line containing one integer indicating the number of iterations.

Sample Input 0
5
2 5
6 5
2 1
7 9
7 2

Sample Output 0
5
8
2
10
6

Sample Input 1
1
5 7

Sample Output 1
7

Explanation 1

The series becomes 5,7,2,5,3,2,1,1,0 after applying the given formula i.e. Next Term = Difference between Previous 2 terms. After that we count the numbers from third term upto zero(last term) ,so the output becomes 7.

Problem 2: M1/P1/L1/24/10pattern

Problem Statement: You need to code for the pattern that is shown in sample for different number of input integers.

Input Format
Input consists of integer.

Constraints
1<=n<=10

Output Format
As in Sample

Sample Input 0
5

Sample Output 0
```
10101
01010
10101
01010
10101
```

Problem 3: M1/P1/L1/20/box spiral

Problem Statement: You are given an integer that is taken as input and you need to print the pattern that moves in spiral making a box like structure the structure might differ in term of spacing.

Input Format
Integer n

Constraints
1<=n<=10

Output Format
Spiral box

Sample Input 0
4

Sample Output 0
```
1  2  3  4
12 13 14  5
11 16 15  6
10  9  8  7
```

Explanation 0
The pattern goes in spiral starting from the largest loop going towards the smaller ones.

Week 4

URL: Week 4

Problem 1: It's all about space

Problem Statement: Unlike usual programmers, Vishal loves strings. Vishal is so fond of strings that he keeps annoying his friends by giving them challenges. One day, one of his friends, Darshan gave him a question.
He asked Vishal to write a program to reduce the given string in such a way that there will be no extra space (i.e., more than 1) between any two words after execution of the program.

Input Format
The first and only line of input expects a string S.

Constraints
1 ≤ length of string S ≤ 1500

Output Format
Output a single string with no extra spaces.

Sample Input 0
   dj j   jf d d d

Sample Output 0
dj j jf d d d

Explanation 0
If any letter contains more than 1 space in its Left or Right side , then extra spaces are omitted leaving only 1 space.
Last letter can have infinite spaces on it's Right side & first letter cannot have any space on it's Left side.

Problem 2: M1/P2/L2/35/sum of left diagonals of a matrix.

Problem Statement: Your task is to take a square matrix as input from the user of rows r and column c. Once this array has been taken as the input, you need to find the summation of the left diagonal of this array .Finding the summation and later printing the array along with the result is the final answer.

Input Format
The first line has the input for rows and columns. The rest lines have the input for the array elements.

Constraints
2<=r, c<=50 -200<=Array[elements]<=500

Output Format
Output should be the array in matrix form with each element with two spaces, and each row in new line. The next line should have the output of the summation. If the dimensions are not correct, then print "Not correct dimensions"

Sample Input 0
2
2
1
5
9
4

Sample Output 0
1 5
9 4
5

Explanation 0
The array is printed first followed by the summation of the left diagonals.(1+4)=5.

Problem 3: Difference Is A Necessary Evil

Problem Statement: A Coding Club named FUNCTOR has given a simple task to its members i.e. to calculate the absolute difference of the most occurring maximal and least occurring minimal numbers from a given list of numbers.

Input Format
The first line of the input contains an integer T denoting the number of test cases.
The first line of each test case contains a integer N denoting number of inputs in the list.
The second line of each test case contains N space separated integers.

Constraints
1 ≤ T ≤ 10
1 ≤ N ≤ 10^5
0 ≤ a[i] ≤ 10^5

Output Format
For each test case, output a single integer denoting the absolute difference.

Sample Input 0
2
2
1 5
5
0 0 0 1 2

Sample Output 0
4
1

Explanation 0
In 1st test case 1 & 5 both are occurring only once so max = 5 & min = 1.
Then absolute difference becomes (5-1) = 4
In 2nd test case 0,1 & 2 occurred 3,1 & 1 time only so most occurring is 0 & least occurring are 1 & 2.
Then min from 1 & 2 is 1 so the absolute difference becomes (1-0) = 1.

Week 5

URL: Week 5

Problem 1: M1/P4/L1/17/Number of Different Strings

Problem Statement: You are given a string as the user enters it, you need to find the number of strings that can be formed from it.

Input Format
The only input line has a string of length n. Each character is between a–z.

Constraints
1≤n≤10^2

Output Format
Print the number of different strings modulo 10^9 +7.

Sample Input 0
aabac

Sample Output 0
15

Explanation 0
There are 15 substrings that can be formed using this given string aabac.

Problem 2: Friends & Balls

Problem Statement: Gaurav and Jai are two friends. One fine day Gaurav asks Jai to solve a problem in which there are N boxes of balls numbered from 1 to N and all the boxes are empty. Everyday Gaurav gives Jai two indices [L,R] and asks him to add 1 to each box from L to R (both inclusive). He repeated this for M number of Days. After M days Jai has a query i.e., What is the number of boxes those contains at least X balls. He has Q such queries .

Input Format
First line contains N - number of ball boxes.
Second line contains M - number of days.
Each of the next M lines consists of two space separated integers L and R.
Followed by integer Q - number of queries.
Each of next Q lines contain a single integer X.

Constraints
1 ≤ N ≤ 10^6
1 ≤ M ≤ 10^6
1 ≤ L ≤ R ≤ N
1 ≤ Q ≤ 10^6
1 ≤ X ≤ N

Output Format
For each query output the number of boxes in new line.

Sample Input 0
7
4
1 3
2 5
1 2
5 6
4
1
7
4
2

Sample Output 0
6
0
0
4

Explanation 0
Let's have a list of ball boxes.
Initially, as shown in the sample test case below we have 7 ball boxes, so let's have an array of 7 integers initialized to 0 (consider 1-based indexing).
array = [0,0,0,0,0,0,0]

After Day 1, array becomes:
array = [1,1,1,0,0,0,0]

After Day 2, array becomes:
array = [1,2,2,1,1,0,0]

After Day 3, array becomes:
array = [2,3,2,1,1,0,0]

After Day 4, array becomes:
array = [2,3,2,1,2,1,0]

Now we have queries on this list:
Query 1: How many boxes have at least 1 ball?
Ans: Ball boxes 1,2,3,4,5 and 6 have at least 1 Ball in them. Hence the output is 6.

Query 2: How many boxes have at least 7 balls?
Ans: We can see that there are no boxes with at least 7 balls. Hence the output is 0.

Query 3: Similar to Query 2.

Query 4: How many boxes have at least 2 balls? Ans: Ball boxes 1,2,3 and 5 have at least 2 coins in them. Hence the output is 4.

Problem 3: Replace with the giant element.

Problem Statement: Given an array of integers, replace every element with the next greatest element (greatest element on the right side) in the array. Since there is no element next to the last element, replace it with -1. For example, if the array is {16, 17, 4, 3, 5, 2}, then it should be modified to {17, 5, 5, 5, 2, -1}.

Input Format
First line of the input contains t, the number of test cases.
Second and third line corresponds to test case 1 (t=1). Fourth and fifth lines correspond to test case 2 (t=2) and so on.
For t=1, the second line contain the value N (size of array elements).
The third line contains N elements of the array
For t=2, the fourth line contain the value N (size of array elements).
The fifth line contains N elements of the array

Constraints
0<N<1001

Output Format
17 5 5 5 2 -1

Sample Input 0
1
6
16 17 4 3 5 2

Sample Output 0
17 5 5 5 2 -1

Explanation 0
Since there is no element next to the last element, replace it with -1. For example, if the array is {16, 17, 4, 3, 5, 2}, then it should be modified to {17, 5, 5, 5, 2, -1}.

Week 6

URL: Week 6

Problem 1: M1/P2/L1/25/Write a program in C to find two elements whose sum is closest to zero.

Problem Statement: For a given array A, you need to code for a problem that requires you to find two such elements that sum upto closest to zero.

Input Format
One integer input for size.
Rest inputs will be of integers.

Constraints
1<=size<=10

Output Format
Only two elements for which the sum is closest to zero

Sample Input 0
5
3
40
6
7
8

Sample Output 0
3 6

Explanation 0
The summation of 3 and 6 is closest to zero.

Problem 2: M1/P4/L1/23/count zero's in l left shift of a given 10 base number

Problem Statement: Suppose you have T attempts and in each attempt, you will be given two numbers A and R for which you have to calculate zeros in resultant of left shift (A<

Input Format
T attempts for each attempt a number will be given and defined by A and R.
for example:
3
60 3
17 2
11 5

Constraints
0<T<100
0<A<256

$0<R<9$

Output Format
Count of each attempt in next line.

Sample Input 0
3
60 3
17 2
11 2

Sample Output 0
5
6
5

Explanation 0
for example: A is 60 and R is 3 here right shift is A<

Problem 3: Indian Formula 1

Problem Statement: As the Formula One Grand Prix was approaching, the officials decided to make the races a little more interesting with a new set of rules. According to the new set of rules, each driver will be given a vehicle with different height and the driver with maximum SIGHT would win the race.
Now, SIGHT of a driver is defined by (X∗P), where
X = number of drivers he can see in front of him + number of drivers he can see behind him
P = position of the driver in a given scenario (index of the driver array is 1−N indexed)
As all the drivers are moving in a straight line, a driver i cannot see beyond another driver j if height of j>= height of driver i.

Input Format
First line of the input contains t, the number of test cases. The 1st line of each test case consists of a single integer n, the number of drivers. Second line contains n space separated integers H[1], H[2], H[3]...H[n] denoting the heights of the drivers 1,2,3....n.

Constraints
0≤t≤50
1≤n≤10^5
0≤H[i]≤10^6

Output Format
Output for each test case should be a single line displaying the index of the winning driver. In case of ties, display the driver with minimum index.

Sample Input 0
2
5
4 1 2 1 4
5
5 1 2 4 1

Sample Output 0
5
4

Explanation 0
As in the 1st test case index no 5 driver can see 4 drivers in front + 0 in back. so value (X*P)= ((4+0)*5) which is highest. so 5 is output.

Week 7

URL: [Week 7]

Problem 1: Ab ki baar kiski Sarkar

Problem Statement: Suppose there are M political parties standing in election. Each party needs to win at least N districts, which just half and more than half of the total districts i.e. H.
We are giving a unique number to each party. And after the election, you have to find the largest party who won more numbers of districts.

Input Format
First line showing T test cases. in next lines, H districts. in next line, H space separated winning party's number.

Constraints
0<T<10
0<H<1000
0<party[H]<10000

Output Format
in each T lines, winning party's number. if there is no party who won half and more than half , than print "void"

Sample Input 0
2
5
2 2 3 1 2
4
1 1 1 3

Sample Output 0
2
1

Explanation 0
in first test case,
in 5 districts, party number "2" won 3 districts, which is more than half of total districts.

Problem 2: Calculate swimming area

Problem Statement: Given an array arr[] of N non-negative integers representing the height of blocks. If width of each block is 1, compute how much swim area can be generated between the blocks during the rainy season.

Input Format
First line contains the no. of blocks. next line having the n elements.

Constraints

3 < N < 10^6
0 < Ai < 10^8

Output Format
only calculated area as integer value.

Sample Input 0
4
4 2 1 6

Sample Output 0
5

Explanation 0
here, blocks are 4, in which for block 2 the trapped area is (4-2)= i.e. 2 similarly, for block 1, the trapped area is 3, and the total trapped area is 2+3=5.

Problem 3: M1/P4/L1/26/ Multiplication of max and min binary numbers

Problem Statement: You take N number of inputs for Binary numbers, Once done with this you need to find the binary with maximum and value and multiply them. Your output should be in Binary.

Input Format
Inputs should be in Binary format.

Constraints
2<=N<=100

Output Format
Output should be in Binary format

Sample Input 0
3
10
100
101

Sample Output 0
1010

Explanation 0
For 3 binary inputs 10,100,101 the multiplication is 10 and in binary is 1010.

Week 8

URL: Week 8

Problem 1: Rotate Array 1

Problem Statement: Write a code to rotate an array circularly with given value to apply left shift.

Input Format
First line of the input contains t, the number of test cases. Each test case contains R rotate value (left shift). Each test case contains N value (size of array elements). Enter all N space separated elements.

Constraints
0<T<1000
0<R<1000
0<N<1000

Output Format
N space separated elements after R rotations.

Sample Input 0
2
3
7
12 43 5 6 7 8 15
2
9
21 43 5 23 87 6 45 90 9

Sample Output 0
6 7 8 15 12 43 5
5 23 87 6 45 90 9 21 43

Explanation 0
In first testcase , rotation value is 3. so given array will be shifted 3 positions in left side. 6 7 8 15 12 43 5.

Problem 2: M1/P4/L1/28/ number is strong number

Problem Statement: You are given N attempts, for which you need to take N numbers as input and then give output if the entered numbers are strong or not.

Input Format
N int of attempts. N number of integer inputs

Constraints
1<=N<=10^6 1<=number<=10^6

Output Format
Output should be Yes or No

Sample Input 0
2
20
145

Sample Output 0
No
Yes

Explanation 0
As there are two attempts 20 is not a strong number. 145 is a strong number.

Problem 3: Find Maximized Calories

Problem Statement: You are at the best toffee shop in the town. You see that there are different varieties of toffees in the shop. Each type of toffee costs P rupees and contains C calories. You being a student have a fixed budget B rupees to spend on these toffees. You want to maximize the total calories you gain by eating them. Note : Any toffee can be eaten only once. Your task is to write a program to output the maximum calories you gain with your budget (B rupees).

Input Format
The first line has an integer T denoting the number of test cases. Then T test cases follow. The first line of each test case has an integer N denoting the number of toffees available. The next line consists of B rupees. Third and fourth lines of T test cases having space separated N elements denoting Cost and calories in respective lines.

Constraints
1<=T<=100 1<=N<=1000

Output Format
For each test case print maximized calories earned in separated lines.

Sample Input 0
2
5
30
10 7 9 13 4
12 14 21 15 9
5
25
10 17 9 13 4
12 14 21 15 9

Sample Output 0
56
42

Explanation 0

In the 1st test case, B rupees that is 30, and we have 5 different toffees available. we need to keep the point in mind that calories must be maximized and cost of these purchased toffees shouldn't exceed the max limit of B rupees.

Week 9

URL: Week 9

Problem 1: Level Ordering

Problem Statement: This problem has been asked several times in many TOP MNC's. First you need to create a binary tree but the insertion of node is slightly different. if a key is divisible by 4 then it will be added at left side. Else it will be added at right side. Now write a function to print the level order traversal for the above generated tree.

Input Format
The first line of input contains an integer T denoting the no of test cases. Then T test cases follow. Each test case contains two lines. The first line of each test case contains an integer N. Then in the next line are N space separated values of the array A[].

Constraints
0<T<50
0<N<1000
0<A[]<10^5

Output Format
For each test case in a new line output will be the level order traversal.

Sample Input 0
3
7
56 77 446 9 8 768 65
9
464 640 368 346 99 789 38 191 334
5
8 10 12 16 21

Sample Output 0
56 8 77 768 446 9 65
464 640 346 368 99 789 38 191 334
8 12 10 16 21

Explanation 0
For the 3rd test case, elements are 8 12 10 16 21. as 8, 12 16 are divisible by 4 so for 1st element is 8, will be the root 12: divisible by 4, so added at left of root 10 not divisible, added at right of root 16 divisible, root->left , but it is not null, again new root is 12 , added left of 12 (sub root) 21 not divisible , root->right , but it is not null, again new root is 10, added right of 10 (sub root of sub tree) 8 12 10 16 21 so level order traversal is 8 12 10 16 21.

Problem 2: M1/P4/L1/27/Number is friendly pair

Problem Statement: For given two number you are required to check if they share common abundancy index or not. friendly numbers are two or more natural numbers with a common abundancy index, the ratio between the sum of divisors of a number and the number itself.

Input Format
Input should be 2 integers N1,N2

Constraints
1<=N1,N2<=10^6

Output Format
Output will be in YES ot NO.

Sample Input 0
6
28

Sample Output 0
YES

Explanation 0
For the input 6 the factors are 1,2,3,6 with summation 12 and abundancy ratio 2. For input 28 the factors are 1,2,4,7,14,28 with summation 56 and abundancy ratio 2.

Problem 3: M1/P1/L1/30/number is a divisor of its right rotation

Problem Statement: Consider the number 142857. We can right-rotate this number by moving the last digit (7) to the front of it, giving us 714285.
It can be verified that 714285=5×142857.
This demonstrates an unusual property of 142857: it is a divisor of its right-rotation.
Consider a function 'find()', such that it returns the last 5 digits of the sum of all integers 'n', 10 < n < 10100, that have this unusual property.

Input Format
There should be no messages (such as, 'Enter the number:').

Constraints
Identifiers for each of the function and variables are case-sensitive, i.e., 'T' and 't' are different.

Output Format
Output should not contain any message (such as, 'Output is:'). It should strictly follow the format provided.

Sample Input 0
2 1342

Sample Output 0
6645

Explanation 0
Here, in this example,

Sample Input = 2 & 1342
Sample Output = 6645.

Week 10

URL: Week 10

Problem 1: M1/P4/L1/29/ Integer as a sum of Two prime Numbers

Problem Statement: For a given set of T attempts, You are required to take input N from the user and for that number your code should be well enough to represent the N as the sum of two positive numbers a and b, count those combinations and that should be the output.

Input Format
There will be integer input for T. And T times integer input N.

Constraints
2<=T<=10^5 0<=N<=10^4

Output Format
There will be only one output for the combinations corresponding to every attempt. and if there is no such number the output should be "not sum of two prime numbers." If number of attempts are greater than 10^5, then print "Too many attempts"

Sample Input 0
1
13

Sample Output 0
1

Explanation 0
For 1 attempt 13, 13 = 2 + 11.

Problem 2: M1/P2/L1/20/ Write a program in C to Print the kth Element in the Array

Problem Statement: For a given array A of size 20, and a given integer k the code should return the array element corresponding to the integer k.

Input Format
Integer to get the size.
Element should be of integer data-type

Constraints
1<size<20

Output Format
Array element

Sample Input 0
5
2

6
8
9
1
2

Sample Output 0
6

Explanation 0
First line has the size of array. Followed by array elements. The last line has the integer corresponding which the array element is to be returned. Here 2nd element of this array is 6

Note: We are not takin g the index 2 to be returned , we require to return the second element that is entered in the array.

Problem 3: M1/P1/L1/3/Valid Triangle

Problem Statement: A triangle is valid if the sum of all the three angles is equal to 180 degrees.

Input Format
The First line contains T, the number of test cases, followed by three integers a, b, c indicating the value of angles in a triangle.

Constraints
1<=T<=1000

Output Format
Print “Valid” if the triangle is valid and “Invalid” if the triangle is invalid.

Sample Input 0
3
60 60 60
40 80 60
50 50 50

Sample Output 0
Valid
Valid
Invalid

Week 11

URL: Week 11

Problem 1: M1/P1/L1/4/Check For Rectangle

Problem Statement: You are given four integers a, b, c and d. Determine if there's a rectangle such that the lengths of its sides are a, b, c and d (in any order).

Input Format
The First line contains T, the number of test cases, followed by four integers a, b, c,d indicating the value of sides of rectangle

Constraints
1<=T<=1000 1<=N<=1000000

Output Format
Print “Valid” if the rectangle is valid and “Invalid” if the rectangle is invalid

Sample Input 0
5
5 10 5 10
3 3 4 4
2 3 3 2
1 1 2 3
5 6 5 6

Sample Output 0
Valid
Valid
Valid
Invalid
Valid

Problem 2: M1/P1/L1/23/ Number pattern

Problem Statement: Your program should print the pattern as given in the Sample as per the input of the integer n from the user.

Input Format
Input should be the integer

Constraints
1<=n<=10

Output Format
Print the pattern as in the Sample

Sample Input 0

4

Sample Output 0

```
   1
  212
 32123
4321234
 32123
  212
   1
```

Problem 3: M1/P6/L1/2/ Count number of times repeated a letter.

Problem Statement: You have to enter a string of length L and you need to count the number of occurrences of a letter which is repeated.

Input Format
Length of the string given by user and defined by L, and the letter c which is repeated in the sentence. for example: 11 l hello world

Constraints
0

Output Format
Output should contain the letter itself. for example Letter l repeated 3 times.

Sample Input 0
11
l
hello world

Sample Output 0
Letter l repeated 3 times.

Faculty CPP Manual

Department of Computer Science and Engineering

Table of contents

Objectives of CPP Program

Guidelines to create Hackerrank Profile

1. *Open* URL: www.hackerrank.com
2. *Click* on **Sign Up & Code** (*for Job Seekers*) as shown in Figure 1.

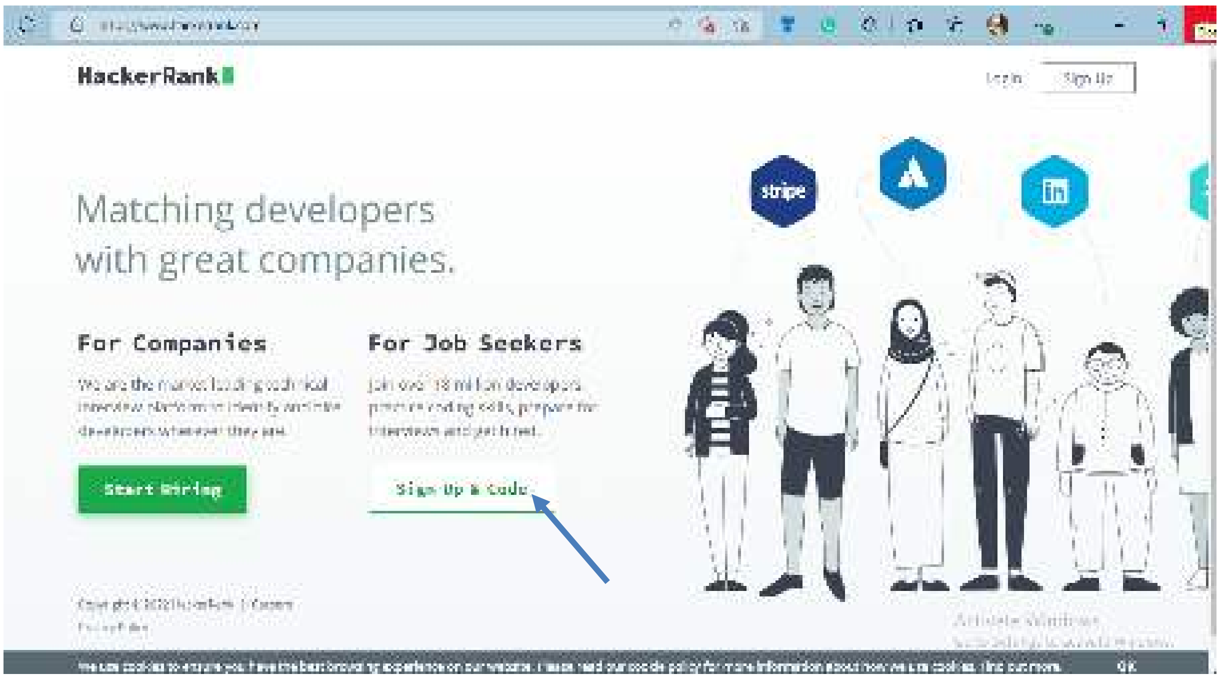


Figure 1. Hackerrank Home page

3. Fill the necessary details required to create an account and *click* on **Create An Account** as shown in Figure 2.

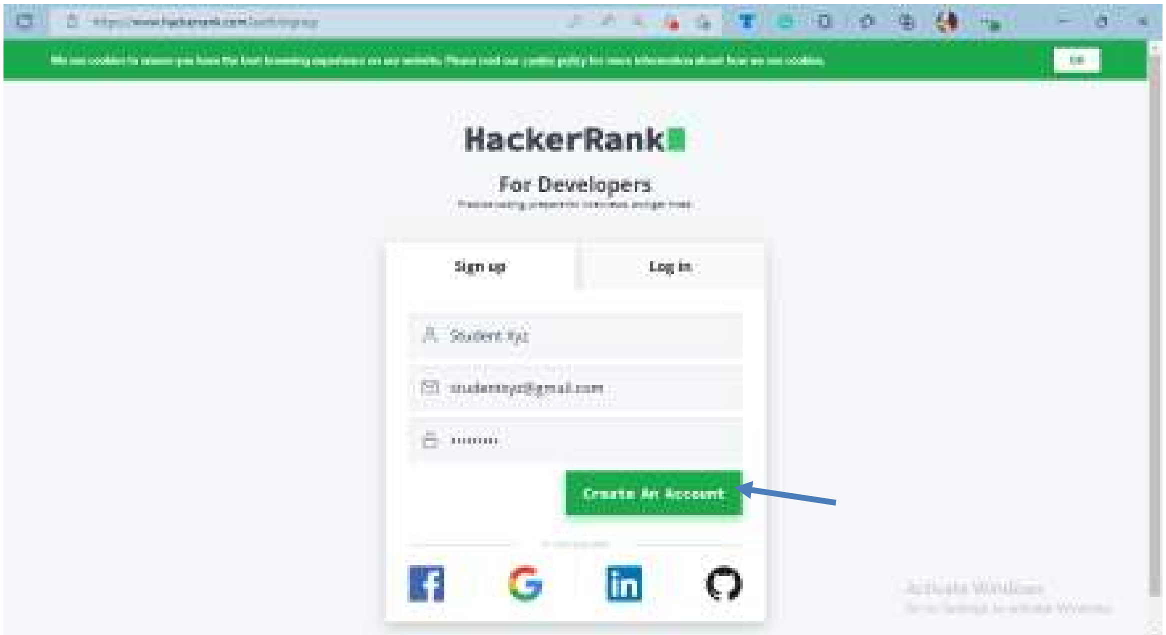


Figure 2. Sign Up page

4. Once sign-up is done, *profile personalization page* appears, *click* on **Learn & Compete with Others** as shown in Figure 3.

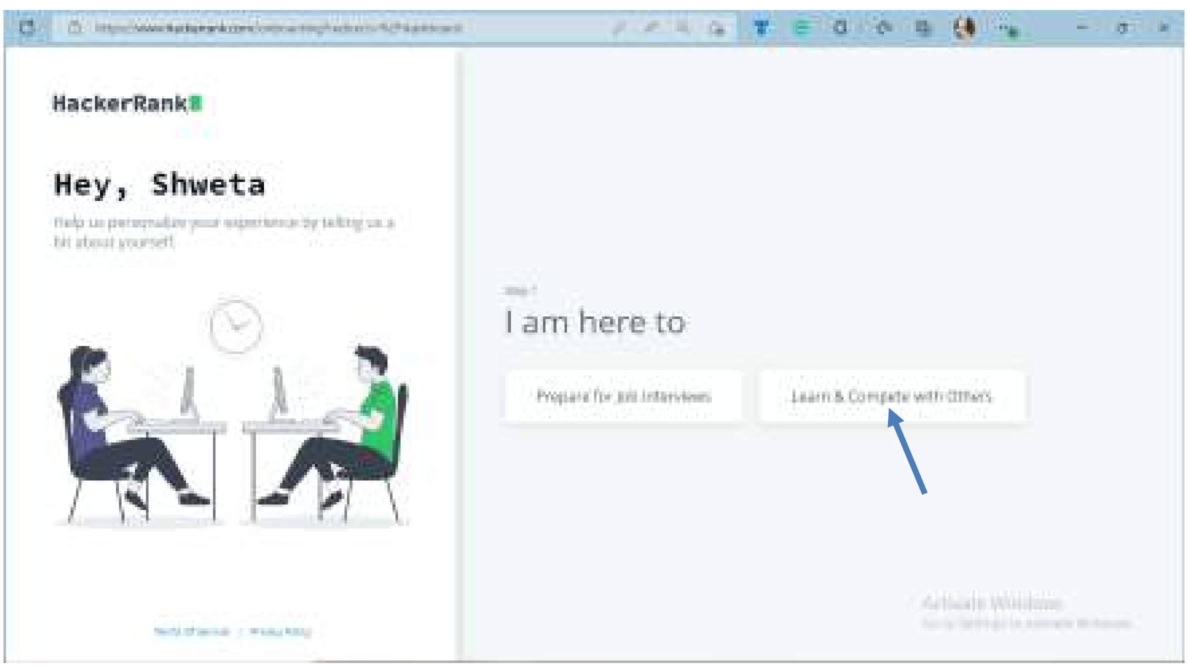


Figure 3. Step-1 (Profile personalization page)

5. *Choose* an appropriate category as shown in Figure 4.

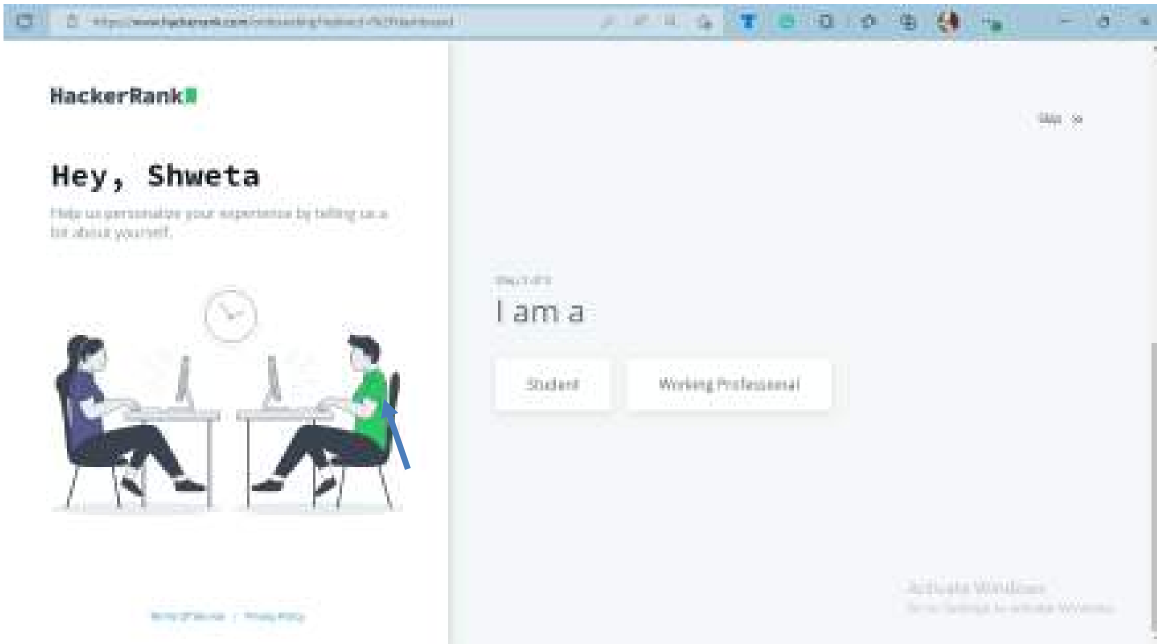


Figure 4. Step-2 (Profile personalization page)

6. *Choose* an year in which you will be graduating, *check the below checkbox* if you feel interested in being contacted through Hackerrank about various job opportunities and *click* on **Let's go** button as shown in Figure 5.

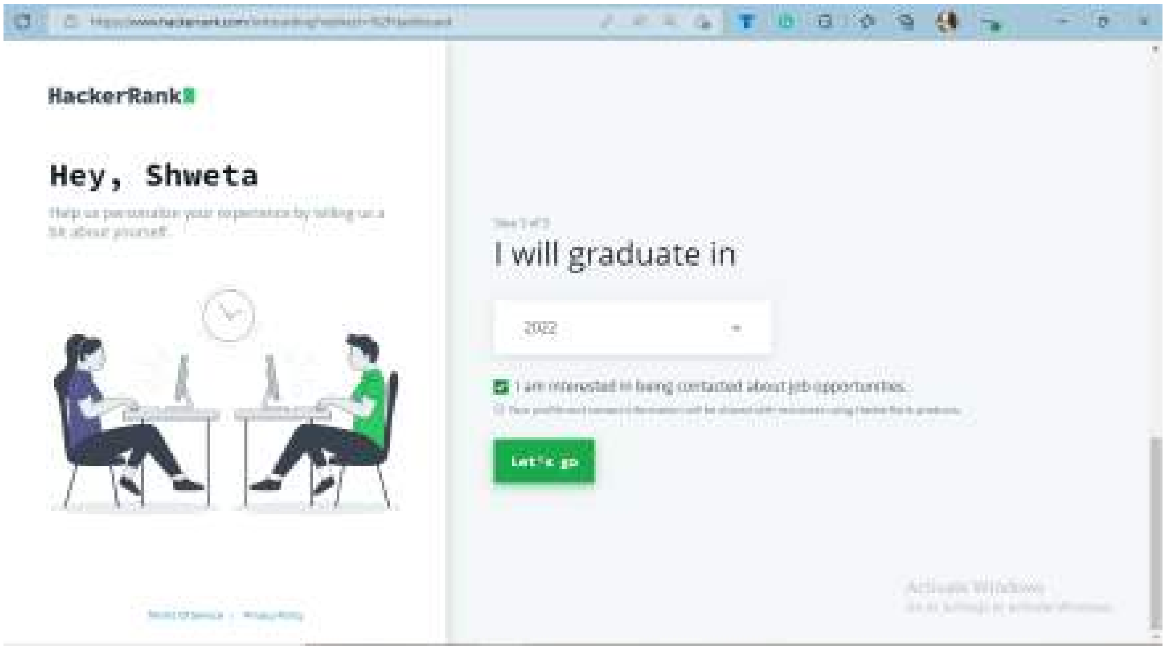


Figure 5. Step-3 (Profile personalization page)

7. Once, all three steps are completed, your profile **Dashboard** will appear as shown in Figure 6, *confirm your email address* by visiting your email account and *clicking* on **Confirm** as shown in Figure 7.

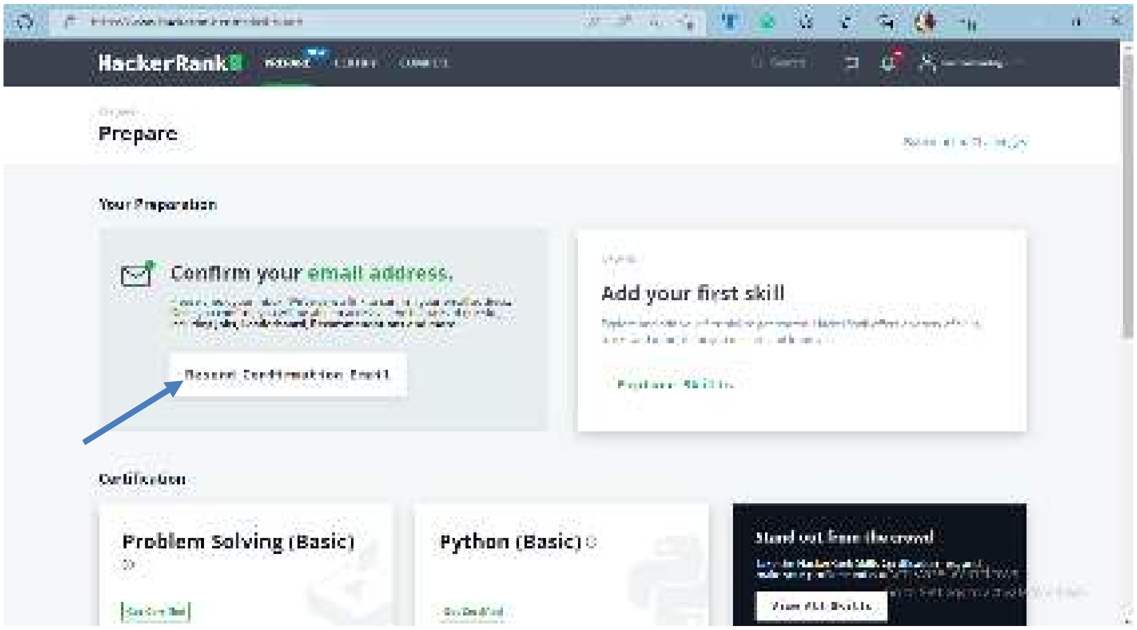


Figure 6. Profile Dashboard

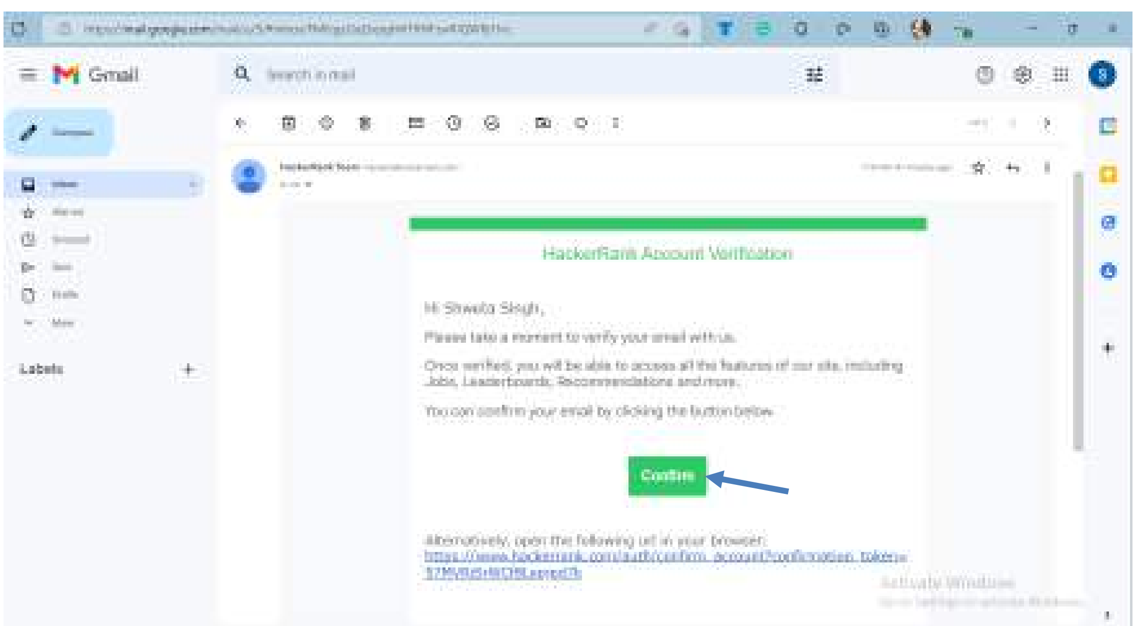


Figure 7. Email Confirmation page on gmail.com

8. By confirming your email address, you will be *directed* to the **Hackerrank Dashboard** page as shown in Figure 8.

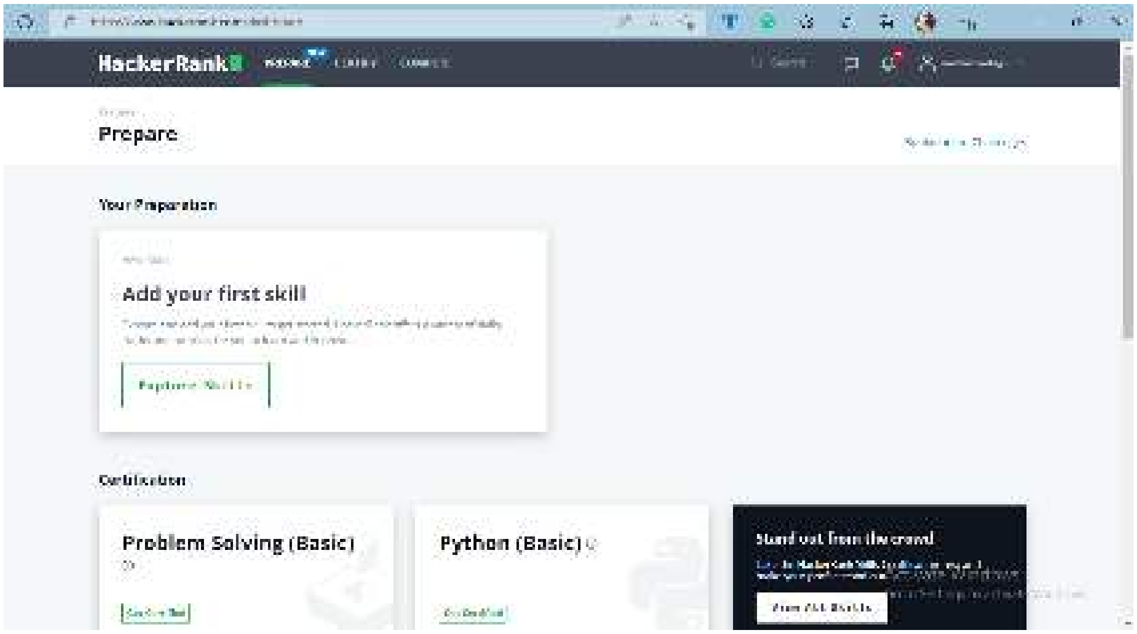


Figure 8. Hackerrank Dashboard page

9. To complete the profile, *click* on the arrow beside your username and *select* **Profile** from the drop-down menu to follow the steps of profile completion as shown in Figure 9, 10 and 11.

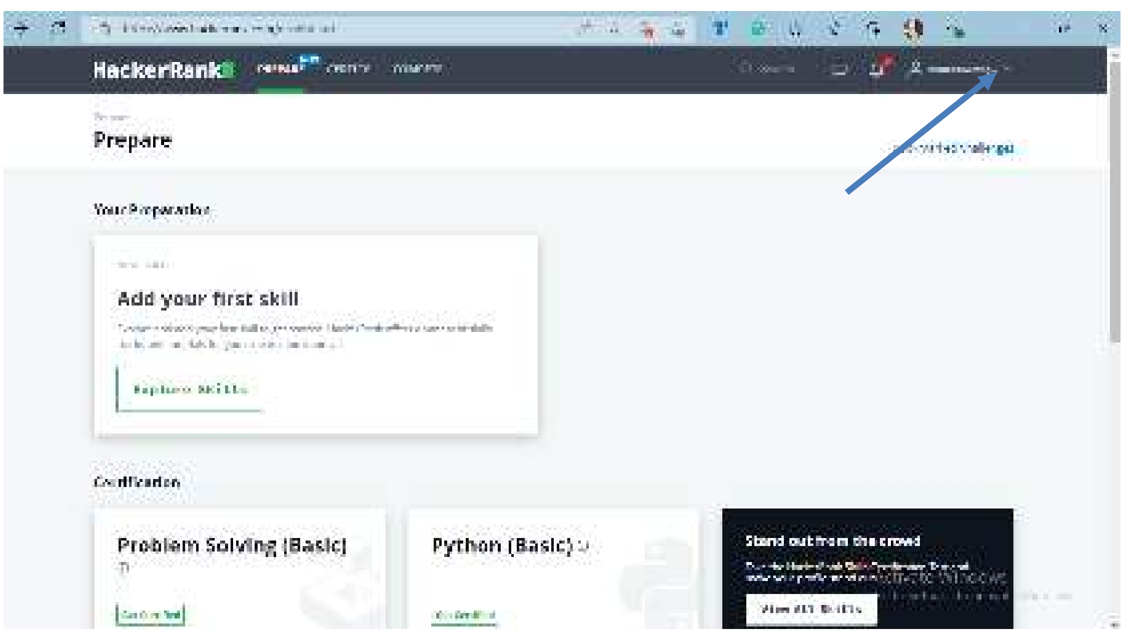


Figure 9. Step-1 Profile Completion

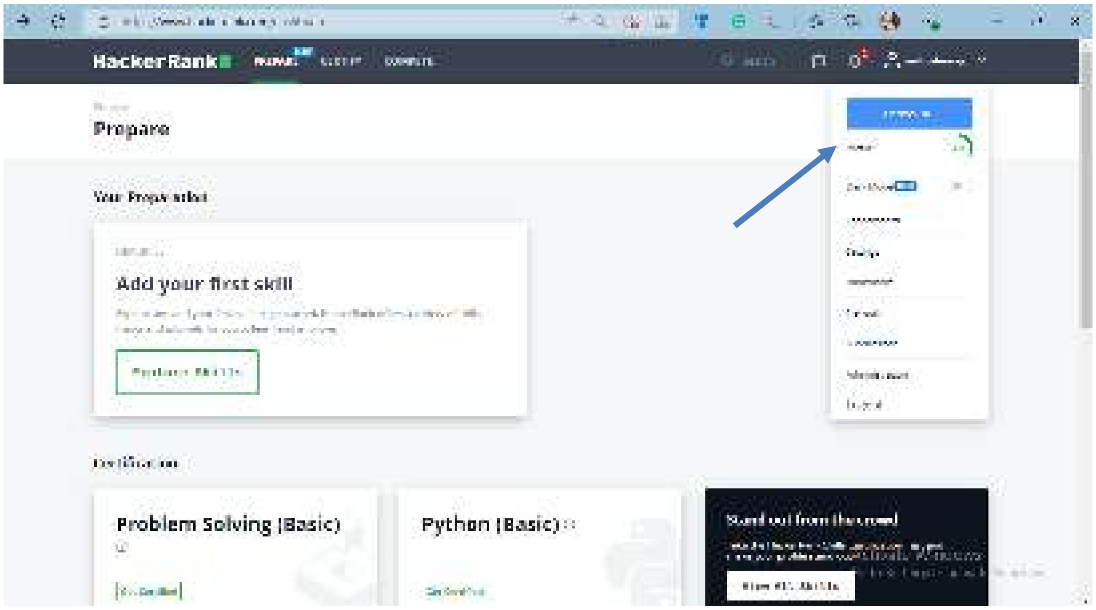


Figure 10. Step-2 Profile Completion

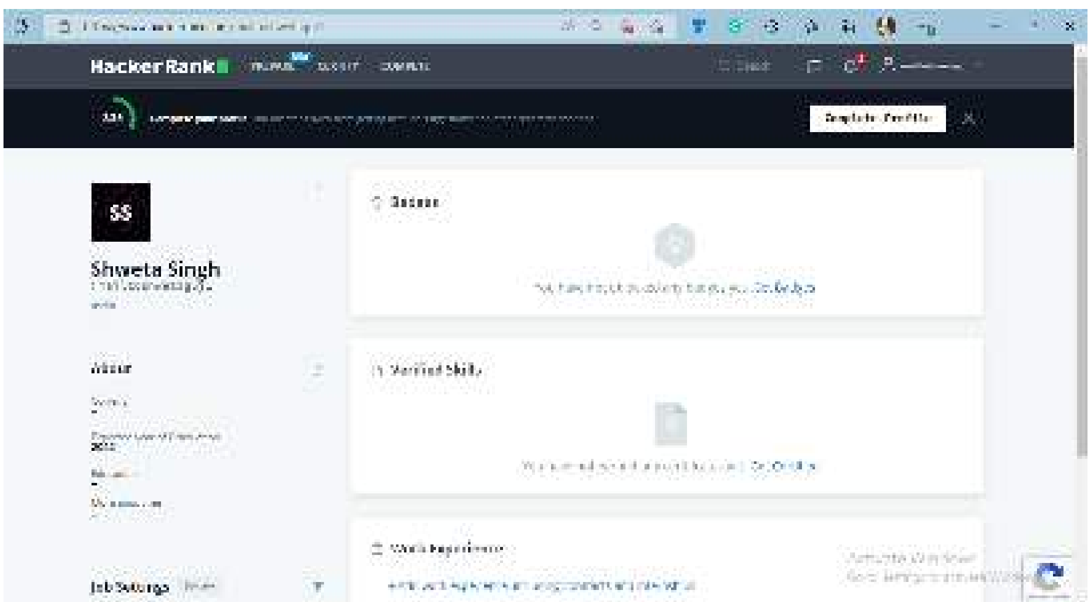


Figure 11. Step-3 Profile Completion

10. At last, to *change your username* and *connect via various social platforms*, *select* **Settings** from the same drop-down menu (as shown in Figure 9) and *update the necessary links* as shown in Figure 12 and 13.

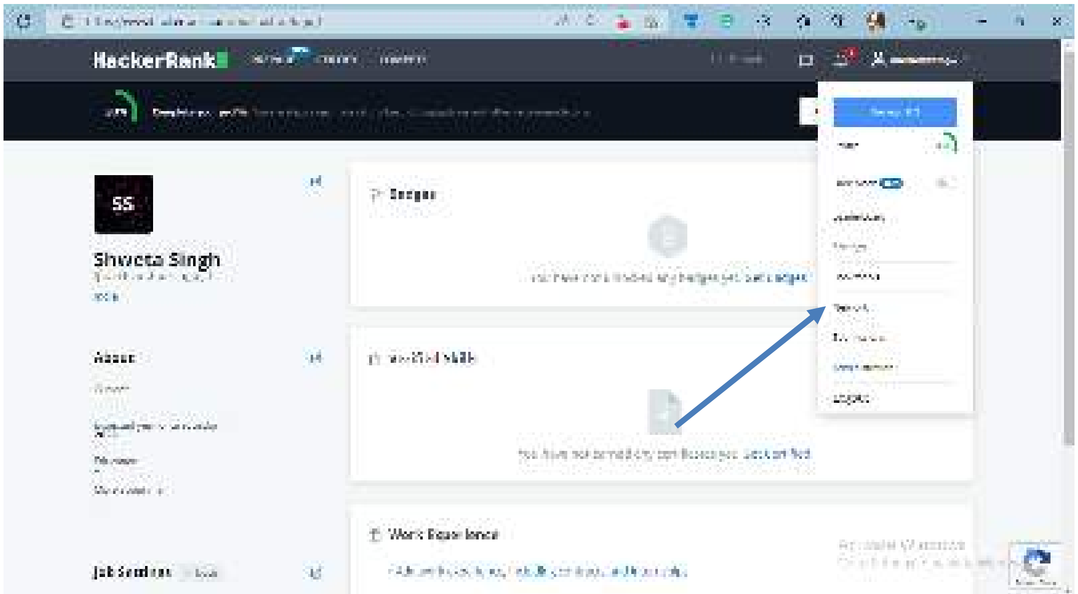


Figure 12. Settings Menu

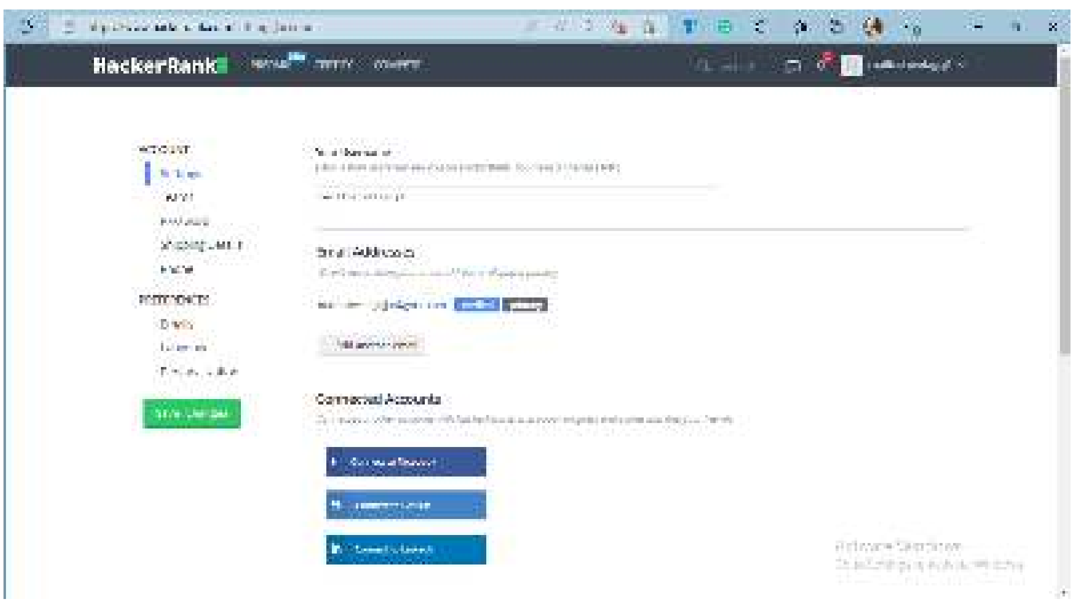


Figure 13. Hackerrank Settings Page

Congratulations! Your Coding Journey has started.

Classwork Problems

Week 1

URL: [Week 1]

Problem 1: M1/P1/L1/8/GCD

Problem Statement: Consider a scenario for 'T' test cases. For each test case 't1', let 'a1' and 'a2' be two positive number and there exist a scenario where 'a1' and 'a2' are divisible by a set of positive numbers S={x1, x2,...xn}.
Compute the greatest number 'x' from the set 'S', such that 'x' divides both 'a1' and 'a2'.

Input Format
The first line contains an integer T, total number of test cases.
Then follow T lines, each line contains an integer a1 and a2.

Constraints
1<=T<=1000
1<=a1,a2<=100000

Output Format
The value of x.
Do not print additional messages.

Sample Input 0
3
5 15
4 24
3 7

Sample Output 0
5
4
1

Explanation 0
Number of test cases = 3
For test case t1, value of a1 = 5, value of a2 = 15
Greatest Common Divisor of a1 and a2 = 5
Similarly for test case t2 and t3.

Solution:
```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>
```

```
int main() {
   int T,a1,a2,i,j ;
   scanf("%d",&T) ;
   for(i=0;i<T;++i)
   {
   scanf("%d",&a1) ;
   scanf("%d",&a2) ;
   {
      for(j=a2;j>0;j--)
      { if (a2%j==0 && a1%j==0)
            break; }
   }
      printf("%d\n",j) ;
   }
   return 0;
}
```

Problem 2: M1/P1/L1/9/Sum of Factorial

Problem Statement: Consider a scenario with 'T' test cases. Hence, for each test case 'ti', let there be any positive number 'n'. Since, factorial of a number 'n' can be computed through the following formula:

$$n! = n \times (n - 1) \times \ldots \times 3 \times 2 \times 1$$

Find a value, named as 'result', by adding all the digits received through a factorial of a number 'n'.

Input Format
The First line contains T, the number of test cases, followed by integer 'n'.

Constraints
1<=T<=1000
1<=n<=1000000

Output Format
Sum of the digits of a computed factorial value.

Sample Input 0
2
5
8

Sample Output 0
3
9

Explanation 0
Number of test cases = 2
For test case t1, value of 'n' = 5
n! = 5*4*3*2*1 = 120
result = 1+2+0 = 3
Similarly, for test cases t2.

Solution:

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {
     int T,n,i,j,sum=0,pr=1;
     scanf("%d",&T) ;
    for(j=0;j<T;++j)
    {
        scanf("%d\n",&n) ;
        for(i=1;i<=n;i++)
        {
            pr = pr * i ;
        }
        while (pr!=0)
        {
        sum = sum + pr%10 ;
           pr=pr/10 ;
        }
         printf("%d\n",sum) ;
         sum = 0 ;
         pr = 1;
    }
     return 0 ;
}
```

Problem 3: M1/P1/L1/Perfect number

Problem Statement: Let, 'n' be any positive number, such that it has a set of positive divisors 'S' = {x1, x2,..., xn}. Hence, a number 'n' is said to be as perfect number, if and only if the sum of all the elements forms the number 'n' itself.

Input Format
A number 'n'.

Constraints
1≤n≤10000

Output Format
Use 'yes', if condition is satisfied, 'no' otherwise.
Do not include additional messages.

Sample Input 0
6

Sample Output 0
yes

Explanation 0

Let 'n' = 6
Factors of 'n', set S = {1, 2, 3}
Sum = 1+2+3 = 6
Hence, 6 is a perfect number.

Solution:

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {
int sum=0,i,n;
scanf("%d",&n);
for(i=1;i<n;i++)
{ if(n%i==0)
   sum=sum+i;
}
if(sum==n)
   printf("yes");
   else
   printf("no");
   return 0;
}
```

Week 2

URL: [Week 2]

Problem 1: M1/P1/L1/5/Sum of Prime

Problem Statement: Consider a scenario with 'T' test cases, where, for each test case 'ti' there exists a number 'n', such that it holds a positive integer value. Hence, a set 'S' is formed out from the series that holds a constraint that:
S=(x1, x2, x3,...)
Every element in the set 'S' is prime in nature.
Hence, 'result' can be concluded as (x1+x2+x3+...)

Input Format
The First line contains T, the number of test cases, followed by integer N.
Do not print additional messages.

Constraints
1<=T<=1000
1<=N<=1000000

Output Format
The value of 'result' for each number 'n'.
Do not print additional messages.

Sample Input 0
3
10
30
50

Sample Output 0
17
129
328

Explanation 0
Number of test cases, T=3
for test case 't1', n = 10
Set S = (2, 3, 5, 7)
Value of result = 2+3+5+7 = 17
Similarly, for test case t2 and t3.

Solution:
```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>
int main()
```

```
{
int x,a;
   scanf("%d",&a);
   for (x=1;x<=a;x++)
   {int arr[1000],count1,count=0,sum=0,i,j,n,flag;
   scanf("%d",&n);
   count1=0;
   for(i=2;i<=n;i++)
   {
      int c=0;
      for(j=1;j<=i;j++)
      {
         if(i%j==0)
         {
            c++;
         }
      }
     if(c==2)
      {
         arr[count1]=i;
         count1+=1;
      }
   }
      //printf("%d",count1);
   for(i=0;i<count1;i++)
      sum+=arr[i];
   printf("%d\n",sum);
  }
}
```

Problem 2: M1/P1/L1/11/Palindrome

Problem Statement: Consider a scenario, where there exists two positive numbers 'n1' and 'n2'. The numbers 'n1' and 'n2' follows certain set of constraints:
They are always positive
They can never hold a value greater than 100
Compute the result 'r' using the formula given below:
r = max (n1*n2) for all 10<=n1,n2<100, such that 'r' is equivalent to its reverse.

Input Format
The maximum value of 'n1' and 'n2'.
Do not print additional messages.

Constraints
10 <= n1, n2 < 100
Both are always positive.

Output Format
The result 'r', if it validates the state, print "Not a valid state" in case of non-valid state.
Do not print additional messages.

Sample Input 0
99
99

Sample Output 0
9009

Explanation 0
Value of 'n1' = 99
Value of 'n2' = 99
Hence, result 'r' = max (n1 * n2) and 'r' is a palindrome.

Value of ‘n1’	Value of ‘n2’	Result ‘r’
10	10	100
10	11	110
10	12	120
.	.	.
.	.	.
.	.	.
99	99	9801

Solution:

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {
   int n1,n2;
    scanf("%d %d" ,&n1,&n2);
    if(n1>=10&&n1<100&&n2>=10&&n2<100)
        {
        int max=0;
        for(int i=10;i<=n1;i++)
            {
            for(int j=10;j<=n2;j++)
                {
                int r=j*i;
                int r1=r,sum=0;
                while(r!=0)
                    {
                    int r2=r%10;
                    sum=sum*10+r2;
                    r=r/10;
                    }
                if(sum==r1)
                    {
                    max=sum;
                    }
```

```
            }
        }
    printf("%d",max);
    }
  else
    {
    printf("Not a valid state");
    }
      return 0;
}
```

Problem 3: M1/P1/L1/1/Play with Numbers

Problem Statement: Write a program to obtain a number N, Such that: Case1: increment it's value by 2 if it is divisible by 11 Case2: decrement it's value by 2 if it is divisible by 4 Case3: else multiply the number by 4

Input Format
The First line contains T, the number of test cases, followed by integer N.

Constraints
1<=T<=1000 1<=N<=100000

Output Format
The new number in new line.

Sample Input 0
3
121
64
15

Sample Output 0
123
62
60

Explanation 0
N=121, it is divisible by 11 so it will fall in case 1 hence updated number is 121+2=123

Solution:

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {
  int n,i,t,p;
  scanf("%d",&t);
  for(i=0;i<t;i++)
  {
```

```
    scanf("%d",&n);

     if(n%11==0)
     {
        p=n+2;
        printf("%d\n",p);

     }
     else if(n%4==0)
     {
        p=n-2;
        printf("%d\n",p);

     }
     else {
        p=n*4;
        printf("%d\n",p);
     }
  }
}
```

Week 3

URL: Week 3

Problem 1: M1/P1/L1/26/complement_of_number

Problem Statement: Suppose, you are given a number 'num', then, there exists a function 'find()' such that it takes an input of type 'integer' and flips its binary representation to compute the 1's complement of a number.

Input Format
num ranges (1,100) There should be no messages (such as, 'Enter the number:').

Constraints
Identifiers for each of the fucntion and variables are case-sensitive, i.e., 'T' and 't' are different.

Output Format
Output should not contain any message (such as, 'Output is:'). It should strictly follow the format provided.

Sample Input 0
5

Sample Output 0
2

Explanation 0
Here, in this example,
Sample Input = 5,
its Binary Representation is 101
1's Complement is 010
Sample Output = 2

Solution:

```
#include<math.h>
#define lli long long int

long int power(int a,int b){
    long int c=1;
    while(b--){
        c*=a;
    }
    return c;
}

void solve(){
    lli n;
    scanf("%lld",&n);
```

```
    int binary[7];
    int i=0;
    while(n>0){
        binary[i]=(n%2==0)?1:0;
        n=n/2;
        i++;
    }
    i--;
    lli no=0;
    while(i>=0){
        no+=(binary[i]*pow(2,i));
        i--;
    }
    printf("%lld",no);
}

int main(){
    lli t=1;
    // scanf("%lld",&t);
    while(t--){
        solve();
    }
    return 0;
}
```

Problem 2: M1/P1/L1/18/count of divisibility

Problem Statement: You have been given 3 integers - l, r and k. Find how many numbers between l and r (both inclusive) are divisible by k. You do not need to print these numbers, you just have to find their count.

Input Format
Integers for l,r,k.

Constraints
1<=l,r,k<=10^5

Output Format
Output should be in the form of integer. In some case if the value of k >l,r print a message "value of k is supposedly larger"

Sample Input 0
4
9
2

Sample Output 0
3

Explanation 0
L=4,r=9,k=2 there are 3 numbers between 4 and 9 that are divisible by 2

Solution:

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {
int i,l,r,k,count=0;
    scanf("%d",&l);
    scanf("%d",&r);
    scanf("%d",&k);
    for(i=l;i<=r;i++){
        if(i%k==0){
            count++;
        }
    }
    printf("%d",count);
    return 0;
}
```

Problem 3: M1/P1/L1/16/Next_term

Problem Statement: Given a number N, find the Nth term in the series 1, 3, 6, 10, 15, 21…

Input Format
The Number n

Constraints
1

Output Format
The nth digit of series.

Sample Input 0
10

Sample Output 0
55

Solution:

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {
  int i,n,r=0;
    scanf("%d\n",&n);
    for(i=1;i<=n;i++)
    {
```

```
        r=r+i;
    }
    printf("%d\n",r);
    /* Enter your code here. Read input from STDIN. Print output to STDOUT */
    return 0;
}
```

Week 4

URL: Week 4

Problem 1: M1/P5/L1/3/Sum of Digits until Single Digit

Problem Statement: We define Digital_Sum of an integer using the following rules:
Given an integer, we need to find the Digital_Sum of the integer.
If x has only 1 digit, then its Digital_Sum is x. Otherwise, the Digital_Sum of x is equal to the Digital_Sum of the sum of the digits of x. For example, the Digital_Sum of 9875 will be calculated as:
Digital_Sum(9875) 9+8+7+5 = 29
Digital_Sum(29) 2 + 9 = 11
Digital_Sum(11) 1 + 1 = 2
Digital_Sum(2) = 2

Input Format
given a number defined by n.

Constraints
1<=n<=100000

Output Format
Sum of digit until unit digit.

Sample Input 0
123

Sample Output 0
6

Solution:
```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {
   int n , sum=0 , num = 0;
   scanf("%d",&n);
   while(n>0)
   {
      sum = sum + (n%10);
      n = n/10;
   }
    while(sum>0)
    {
       num = num + (sum%10);
       sum = sum/10;
    }
```

```
    printf("%d",num);
    return 0;
}
```

Problem 2: M1/P6/L2/6/ find the sum of digit in a string.

Problem Statement: You have to enter a string of length L which may consists some digit as well. You need to find that digits and print the sum of all digits present in the string.

Input Format
You have to enter a string of length L
for example:
50
hello1 world23 its a pr5ogr4m6

Constraints
0

Output Format
Output should display the digits present in the string first and then the sum of digits in next line.

Sample Input 0
30
hello1 world23 its a pr5ogr4m6

Sample Output 0
The digits present in the string is 123546
The sum of digits is 21

Solution:

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {
int n;
    scanf("%d",&n);
    char a[n],b=0;
    for(int i=0;i<=n;i++){
        scanf("%c",&a[i]);
    }
    printf("The digits present in the string is ");
    for(int i=0;i<=n;i++){
        // printf("%c",a[i]);
        if(a[i]>=48 && a[i]<=57){
            printf("%c",a[i]);
            int z=a[i]-'0';
                b=b+z;
        }
    }
```

```
    printf("\nThe sum of digits is %d",b);
    return 0;
}
```

Problem 3: Armstrong Numbers 11

Problem Statement: An Armstrong number is a number that is sum of its own digits each raised to the power of the number of digits. Write a program to print all the Armstrong numbers in given range

Input Format

First line consists of an integer T which is the number of testcases.And then there will be T line-separated numbers each line consisting of upper bound and lower bound to print all the Armstrong numbers.

Constraints

1

Output Format

T number of line separted armstrong number,each line consisting of armstrong number(s)

Sample Input 0

1
100 999

Sample Output 0

153 370 371 407

Solution:

```
int main() {
    int t,a,b,d=0,i,j,n,sum=0,s,c,e;
    scanf("%d",&t);
    for(i=0;i<t;i++)
    {
        scanf("%d%d",&a,&b);
        if(a>b)
        {
            c=a;
            e=b;
        }
        else{
            c=b;
            e=a;
        }
        for(j=e;j<=c;j++)
        {
            n=j;
            while(n!=0)
            {
                n=n/10;
                d++;
            }
            n=j;
```

```
        while(n!=0)
        {
           s=n%10;
           sum=sum+pow(s,d);
           n=n/10;
        }
        if(sum==j)
           printf("%d ",j);
        sum=0;
        d=0;
     }
  printf("\n");
  }

  /* Enter your code here. Read input from STDIN. Print output to STDOUT */
  return 0;
}
```

Week 5

URL: Week 5

Problem 1: M1/P2/L2/33/Write a program in C to delete an element at desired position from an array

Problem Statement: You will be given an array of size 6 and we have to delete an element from second position.

Input Format
1 2 3 4 5 6

Constraints
N=6

Output Format
1 2 4 5 6

Sample Input 0
1 2 3 4 5 6

Sample Output 0
1 2 4 5 6

Solution:
```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {
int arr[6];
    int n=6,i;
    for(i=0;i<n;i++)
    scanf("%d",&arr[i]);
    for(i=2;i<n;i++)
        arr[i]=arr[i+1];
     for(i=0;i<n-1;i++)
    printf("%d ",arr[i]);
    return 0;
}
```

Problem 2: M1/P2/L1/6/print all unique elements in an array

Problem Statement: You will be given an array for which you have to print all unique elements in that array

Input Format

10 numbers separated by blank space

Constraints
Total elements N=10
Each element can vary between 1 to 10 (both inclusive)

Output Format
Unique numbers separated by blank space

Sample Input 0
3 4 5 5 6 7 7 6 7 6

Sample Output 0
3 4

Solution:
```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {
int arr[100],c,j,i;
   //scanf("%d",&n);
   for(i=0 ; i<10 ; i++)
   {
      scanf("%d",&arr[i]);
   }
     for(i=0;i<10;i++)
   { c=0;
      for(j=0;j<10;j++)
   {
          if (i!=j)
          { if(arr[i]==arr[j])
      {
         c++;
      }
          }
      }
          if(c==0)
    printf("%d ",arr[i]);
   }
   //printf()
   return 0;
}
```

Problem 3: M1/P2/L1/8/count the frequency of each element of an array

Problem Statement: In this program, we need to count the occurrence of each unique element present in the array. One of the approach to resolve this problem is to maintain one array to store the counts of each

element of the array. Loop through the array and count the occurrence of each element and store it in another array.
1 2 4 7 2 2 2 4 1
In the above array, 1 has appeared 2 times, so, the frequency of 1 is 2. Similarly, 2 has appeared 4 times. The frequency of 2 is 4 and so on.

Input Format
1 2 4 7 2 2 2 4 1

Constraints
Total number of elements N=9
Each element lie between 1 to 99

Output Format
1 2
2 4
4 2
7 1

Sample Input 0
1 2 4 7 2 2 2 4 1

Sample Output 0
1 2
2 4
4 2
7 1

Explanation 0
Frequency of each number in the input

Solution:
```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main()
{
    int arr[9];
    for (int i=0; i<9; i++)
    {
        scanf("%d",&arr[i]);
    }
    int f[9];
    int v = -1;

    for (int i=0; i<9; i++)
    {
        int count = 1;
        for(int j = i+1; j < 9; j++)
```

```
        {
            if(arr[i] == arr[j])
            {
                count++;
                f[j] = v;
            }
        }
        if(f[i] != v)
        f[i] = count;
    }
        for(int i = 0; i < 9; i++){
        if(f[i] != v){
            printf("%d", arr[i]);
            printf(" %d\n", f[i]);
        }
    }
    return 0;
}
```

Problem 4: M1/P2/L1/16/find the largest and second largest element in an array.

Problem Statement: You will be given an array of size 6 for which you have to find the largest and second largest element in an array.

Input Format
1 2 5 6 3 2 2 3 4 6 7 8

Constraints
N=6

Output Format
6 5 8 7

Sample Input 0
1 2 5 6 3 2

Sample Output 0
6
5

Solution:

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {
  int a[6],i=0,temp=0;
    for(i=0;i<6;i++){
        scanf("%d", &a[i]);
    }
```

```
    int max=a[0];
    for(i=1;i<6;i++){
        if(a[i]>a[i-1]){
            temp=max;
            max=a[i];
        }
    }
    printf("%d\n",max);
    printf("%d", temp);
    return 0;
}
```

Problem 5: Crossword Puzzle 1

Problem Statement: You need to play a crossword puzzle.

You are given a 2D square matrix of characters. Your task is to find whether a given word can be found in the matrix by going from left to right or from top to bottom.
Example,
[['F', 'A', 'C', 'E'],
 ['A', 'B', 'C', 'D'],
 ['C', 'A', 'O', 'B'],
 ['T', 'N', 'S', 'O']]
If the required word is "FACE", then it can be found in row 0.
If the required word is "BAN", then it can be found in col 1 (starting from row 1).
If the required word is "FACT", then it can be found in col 0.

Input Format
The first line contains size of the square matrix N
The next N lines contains element of the 2D square matrix.
The last line contains the target word that need to be searched.

Constraints
The matrix is square

Output Format
The output is either True or False depending on whether the word is present in matrix or not.

Sample Input 0
4
FACE
ABCD
CAOB
TNSO
BAN

Sample Output 0
True

Sample Input 1
4

FACE
ABCD
CAOB
TNSO
FACE

Sample Output 1
True

Solution:

```
#include <stdio.h>
#include<string.h>

void crossword_puzzle(char matrix[100][100], int N, char target_word[100])
{
   int i, j, k, index;
   char str[100];
   for(i=0; i<N; i+=1)
   {
      for(j=0; j<N; j+=1)
      {
         index = 0;
         //find word in the same row
         for(k=j; k< N; k=k+1)
         {
            str[index] = matrix[i][k];
            str[index+1] = '\0';//add null character
            if(strcmp(str, target_word) == 0)
            {
               printf("True");
               return;
            }
            index += 1;
         }
index = 0;
         //find word in the same column
         for(k=i; k< N; k=k+1)
         {
          str[index] = matrix[k][j];
            str[index+1] = '\0';//add null character
            if(strcmp(str, target_word) == 0)
            {
               printf("True");
               return;
            }
            index += 1;
         }
      }
   }
   printf("False");
}
```

```
int main() {
  // Write C code here
  int i, j, N;
  char matrix[100][100], target_word[100];
  scanf("%d", &N);
  // Input matrixa
  for(i=0; i<N; i+=1)//num of rows
  {
    for(j=0; j<N; j+=1)//num of columns
    {
      scanf(" %c", &matrix[i][j]);
    }
  }
  scanf("%s", target_word);
  crossword_puzzle(matrix, N, target_word);
  return 0;
  }
```

Week 6

URL: Week 6

Problem 1: M1/P2/L2/22/find transpose of a given matrix

Problem Statement: Transpose of a matrix is obtained by changing rows to columns and columns to rows. In other words, transpose of A[][] is obtained by changing A[i][j] to A[j][i]. Consider the size of matrix N*N. Where N is 3 (fixed)

Input Format
1 2 3 4 5 6 7 8 9 // 1 2 3 elements of first row, 4 5 6 elements of second row, 7 8 9 elements of third row
1 4 0 -5 2 7 //1 4 0 elements of first row, -5 2 7 elements of second row

Constraints
0<N<100

Output Format
1 4 7 2 5 8 3 6 9 // 1 4 7 elements of first row, 2 5 8 elements of second row, 3 6 9 elements of third row 1 -5 4 2 0 7 //1 -5 elements of first row, 4 2elements of second row, 0 7 elements of third row

Sample Input 0
1 2 3 4 5 6 7 8 9

Sample Output 0
1 4 7 2 5 8 3 6 9

Solution:

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {
int arr[3][3],i,j;
for(i=0;i<3;i++)
{
for (j=0;j<3;j++)
{
scanf("%d ",&arr[i][j]);
}
}
for(i=0;i<3;i++)
{
for (j=0;j<3;j++)
{
printf("%d ",arr[j][i]);
}
}
```

```
return 0;
}
```

Problem 2: M1/P2/L1/3/sum of all elements of the array

Problem Statement: Suppose you have A attempts and in each attempt you will be given an array for which you have to find the sum of all elements of array.

Input Format
Enter the size of array: 5 1 5 4 3 2

Constraints
0<N<100

Output Format
15

Sample Input 0
1
2
3
3
2

Sample Output 0
11

Solution:

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {
int sum=0;
   int arr[5];
   for(int i=0;i<5;i++)
   {
      scanf("%d",&arr[i]);
   }
   for(int j=0;j<5;j++)
   {
      sum = sum + arr[j];
   }
   printf("%d",sum);

   return 0;
}
```

Problem 3: M1/P2/L1/22/Write a program in C to rotate an array by N positions

Problem Statement: You are given an array A of size size, Wherein you are supposed to code a program that allows the user to shift the array by n positions either to left or to right.

Input Format

The first line shall include the aray size , that will be in positive integer. Second line shall include the number of rotations. The third line and follwing them should have the array elements.

Constraints
1<size<=10

Output Format
The output should be in two lines 1st: the original array 2nd: the shifted array

Sample Input 0
5
2
10
20
30
40
50

Sample Output 0
10 20 30 40 50
40 50 10 20 30

Explanation 0
5 being the size of the array, 2 being the number of rotations To the RIGHT (you are required to do the same). Mind the space according to the sample output.

Solution:
```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {
    //Initialize array
    int arr[10],k=0 ,Size=0 ;
     scanf("%d",&Size);
      int n =0 ;
    scanf("%d",&n);
     for(k = 0; k < Size; k++)
  {
      scanf("%d", &arr[k]);
  }
    //Calculate length of array arr
    int length = Size   ;
    //n determine the number of times an array should be rotated
    //Displays original array
```

```
    for (int i = 0; i < length; i++) {
        printf("%d ", arr[i]);
    }
    //Rotate the given array by n times toward right
    for(int i = 0; i < n; i++){
        int j, last;
        //Stores the last element of the array
        last = arr[length-1];
        for(j = length-1; j > 0; j--){
            //Shift element of array by one
            arr[j] = arr[j-1];
        }
        //Last element of array will be added to the start of array.
        arr[0] = last;
    }
    printf("\n");
    //Displays resulting array after rotation
    for(int i = 0; i< length; i++){
        printf("%d ", arr[i]);
    }
    return 0;
}
```

Problem 4: M1/P2/L1/24/Write a program in C to find the two repeating elements in a given array.

Problem Statement: You are given an array A[] of maximum size 10, you need to write a program that will enable you to find the duplication of elements in that array.

Input Format
Integer for size.

Constraints
1<=size<=10
Each array element will vary between 1 to 10 (both inclusive)

Output Format
Output will be both the numbers of duplication.

Sample Input 0
5
1
2
3
4
4

Sample Output 0
4 4

Explanation 0
For given size 5 , elements are entered. 4 occurs in duplication in that array. Hence 4 4

Solution:

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {
    int terms,arr[10];
    scanf("%d",&terms);
    for(int i=0; i<terms;i++)
    {
        scanf("%d",&arr[i]);
    }
    for(int j=0; j<terms;j++)
    {
        for(int i=j+1;i<terms;i++)
        {
            if(arr[j]==arr[i])
            {
                printf("%d %d",arr[j],arr[i]);
            }
        }
    }
    /* Enter your code here. Read input from STDIN. Print output to STDOUT */
    return 0;
}
```

Problem 5: Overlapping Series

Problem Statement: You have N series. You are given starting point ai and ending point bi for each series.
You need to identify and remove the redundant series i.e., the series which is part of a larger series.

Input Format
The first line contains N : The number of series
The next N lines contains the starting point ai and ending point bi of each series

Constraints
2 <= N <= 100
1 <= ai , bi <= 10000

Output Format
The starting and ending point of non-overlapping series in the order they were entered

Sample Input 0
4
1 3
5 8
4 10
20 25

Sample Output 0
1 3
4 10
20 25

Explanation 0
The series (5, 8) is entirely overlapped by a larger series (4, 10) so it is removed

Solution:

```
 #include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main()
{
    int n,**ptr,**arr,size=0;
    scanf(" %d",&n);
    arr=(int**)malloc(sizeof(int*)*n);
    ptr=(int**)malloc(sizeof(int*)*n);
    for(int i=0;i<n;i++){
        ptr[i]=(int*)malloc(sizeof(int)*2);
        arr[i]=(int*)malloc(sizeof(int)*2);
        scanf(" %d%d",&ptr[i][0],&ptr[i][1]);
    }
    for(int i=0;i<n;i++){
        for(int j=0;j<n;j++){
            if(i==j)continue;
            if(ptr[i][0]<=ptr[j][0] && ptr[i][1]>=ptr[j][1]){
                arr[size][0]=ptr[j][0];
                arr[size][1]=ptr[j][1];
                size++;
            }
        }
    }
    int fg=0;
    for(int i=0;i<n;i++){
        fg=0;
        for(int j=0;j<size;j++){
            if(ptr[i][0]==arr[j][0] && ptr[i][1]==arr[j][1]){
                fg=1;
                break;
            }
        }
        if(fg==0){
            printf("%d %d\n",ptr[i][0],ptr[i][1]);
        }
    }
    return 0;
}
```

Problem 6: Plus One 6

Problem Statement: You are given a large integer represented as an integer array digits, where each digits[i] is the ith digit of the integer. The digits are ordered from most significant to least significant in left-to-right order. The large integer does not contain any leading 0's.
Increment the large integer by one and return the resulting array of digits.

Input Format
an integer number N

Constraints
0<N<10000
0<M<10
where M is the number of digits in a number.

Output Format
incremented array

Sample Input 0
123

Sample Output 0
124

Explanation 0
The array represents the integer 123. Incrementing by one gives 123 + 1 = 124

Solution:
```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main()
{
   int a[1] ;

   scanf("%d",&a[0]) ;
   printf("%d",a[0]+1) ;
   return 0;
}
```

Week 7

URL: Week 7

Problem 1: M1/P4/L1/1/Sum of First and Last Digits of a Number

Problem Statement: Suppose you have T attempts and in each attempt, you will be given numbers for which you have to find the sum of first and last digit of a number.

Input Format
T attempts for each attempt a number will be given and defined by N.
for example:
3
5656
65657676766787676
65745356789

Constraints
0<T<100
0<N<10^18

Output Format
sum of each attempt in next line.

Sample Input 0
3
2365
6547
6987

Sample Output 0
7
13
13

Explanation 0
for example: in 1st attempt the number is 2365, and first and last digit summation is 2+5=7.

Solution:

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

  int main() {
    int T,i,j,count=0 , size;
    long long int n,n1 ;
    scanf("%d",&T) ;
    for(i=0;i<T;++i)
```

```
    {
      scanf("%lld",&n) ; //value of n
      n1 = n ;
      count = 0 ;
      while(n!=0)
      {
         count++ ;
         n = n/10 ;
      }
      size = count ;
     int arr[size] ;
      for(j=0;j<size;++j)
      {
         arr[j]=(n1%10) ;
         n1 = n1/10 ;
      }
      printf("%d\n",(arr[0]+arr[size-1])) ;
    }
   return 0;
}
```

Problem 2: M1/P4/L1/3/Sum odd digits

Problem Statement: Suppose you have T attempts and in each attempt, you will be given a number for which you have to find the sum of odd positional digits of that number.

Input Format
T attempts, for each attempt a number will be given and defined by N. for example:
3
5656
65657676766787676
65745356789

Constraints
0<T<100
0<N<10^18

Output Format
sum of each attempt in next line.

Sample Input 0
3
2365
65471
6987

Sample Output 0
8
11
14

Solution:

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {
  long int n,d,sum,T,rev;
  scanf("%ld",&T);
  while(T>0&&T<100)
  {
    scanf("%ld",&n);
    sum=0;
    d=0;
    rev=0;
    while(n>0 && n<pow(10,18))
    {
      rev=rev*10+(n%10);
      n=n/10;
    }
    while(rev>0)
    {
      d++;
      if((d%2)!=0) sum=sum+(rev%10);
      rev=rev/10;
    }
    printf("%ld\n",sum);
    T--;
  }
  return 0;
}
```

Problem 3: M1/P4/L1/4/sum of modulo K of first N natural number

Problem Statement: Suppose you have T attempts and in each attempt, you will be given two numbers K and N to find the sum of modulo K of first N natural numbers.

Input Format

T attempts each attempt have a two numbers K and N for which have to find the sum of modulo K of first N natural numbers.
for example:
3
40 245
12 356
34 4525464

Constraints

0<T<100
0<K<100
0<N<10^9

Output Format
each attempt output space separated on next line.

Sample Input 0
2
40 245
12 356

Sample Output 0
4695
1950

Explanation 0
Input : N = 10 and K = 2.
Output : 5
Sum = 1%2 + 2%2 + 3%2 + 4%2 + 5%2 + 6%2 +
 7%2 + 8%2 + 9%2 + 10%2
 = 1 + 0 + 1 + 0 + 1 + 0 + 1 + 0 + 1 + 0
 = 5
similarly; if K 12 and N=356 then answer is 1950.

Solution:
```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {
  int t,n,k,i,j,sum;
  scanf("%d",&t);
  for(i=0;i<t;i++)
  {
    scanf("%d %d",&k,&n);
    sum=0;
    for(j=1;j<=n;j++)
    {
      sum+=(j%k);
    }
    printf("%d\n",sum);
  }
  return 0;
}
```

Problem 4: M1/P4/L1/5/Count the Fibs?

Problem Statement: Let us define a Fibonnaci numbers as:
f1 = 1
f2 = 2
fn = fn-1 + fn-2
You are given two numbers x and y, find the Fibonnaci numbers that lie in the range (x,y).
Note the first Fibonnaci number is assumed to be 1

Input Format

There are T attempts. For each attempt you are given two non-negative inputs x and y. For example:
3
10 100
5 20
1 1000

Constraints

x <= y <= 1010

Output Format

For each attempt, the output is displayed on a single line.

Sample Input 0

2
10 100
1234567890 9876543210

Sample Output 0

5
4

Solution:

```
 #include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {
    int t;
    scanf("%d",&t);
    while(t--){
        long int a = 0,b = 1,c = 0;
        int sum = 0;
        long int m,n;
        scanf("%ld%ld",&m,&n);
        while(c<=n){
            c = a+b;
            a = b;
            b = c;
            // printf("%d\n",c);
            if(c > m && c < n)sum ++;
        }
        printf("%d\n",sum);
    }
    return 0;
}
```

Problem 5: M1/P4/L1/2/Carmichael Numbers

Problem Statement: A number n is said to be a Carmichael number if it satisfies the following modular arithmetic condition:

power(b, n-1) MOD n = 1,
for all b ranging from 1 to n such that b and
n are relatively prime, i.e, gcd(b, n) = 1

Suppose you have T attempts and in each attempt, you will be given N numbers to find the Carmicheal number of each.

Input Format

T attempts each attempt have a number N for which we have to find whether it is Carmichael number or not.
for example:
3
561
8
2365

Constraints

0<T<100
0<N<10^9

Output Format

each attempt output space separated on next line.

Sample Input 0

3
561
8
2365

Sample Output 0

true
false
false

Explanation 0

Here Attempts are 3,
Input : n = 8 Output : false Explanation : 8 is not a Carmichael number because 3 is relatively prime to 8 and (3^8-1) % 8 = 2187 % 8 is not 1.
Input : n = 561 Output : true

Solution:
```
#include<stdio.h>
int gcd(int a,int b){
    if(a<b)
        return gcd(b,a);
    if(a%b==0)
        return b;
    return gcd(b,a%b);
}
```

```
int power(int a,int b,int n){
   int p=1,i;
   for(i=1;i<=b;i++){
      p=(p*a)%n;
   }
   return p;
}

int isCarmichaelNumber(int n)
{
   for(int b=2;b<n;b++)
   {
      if(gcd(b,n)==1)
         if(power(b,n-1,n)!=1)
            return 0;
   }
   return 1;
}

int main(){
   int T,m=0,n;
   scanf("%d",&T);
   while(T!=m){
      scanf("%d",&n);
      if(isCarmichaelNumber(n)==1)
         printf("true\n");
      else
         printf("false\n");
   m++;
   }
   return 0;
 }
```

Problem 6: M1/P4/L1/7/ calculate nCr

Problem Statement: Suppose you have T attempts and in each attempt, you will be given two numbers for which you have to find the nCr.
Logic: nCr = (n!) / (r! * (n-r)!)

Input Format
T attempts for each attempt two numbers space separated will be given. for example: 3 5 2 3 1

Constraints
0<T<100
0<n<1000
0<r<1000

Output Format
each attempt nCr output on next line.

Sample Input 0

2
5 2
3 1

Sample Output 0
10
3

Explanation 0
here Attempts are T=3 for first attempt, n=5 and r=2
then (5C2)=10

Solution:

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int fact(int a) ;
int main() {
    int t,n,r,i;
    scanf("%d",&t) ;
    for(i=0;i<t;++i)
    {
        scanf("%d",&n) ;
        scanf("%d",&r) ;
    printf("%d\n", (fact(n))/(fact(r)*fact(n-r))) ;
    }
    return 0;
}

 int fact(int a)
{
     int i , prod = 1 ;
   for(i=1;i<=a;++i)
   {
       prod = prod*i ;
   }
     return prod ;
}
```

Week 8

URL: Week 8

Problem 1: M1/P4/L1/8/ Square root of a number by Repeated Subtraction method

Problem Statement: Suppose you have T attempts and in each attempt, you will be given a number for which you have to find the Square root of a number by Repeated Subtraction method.
Logic:
N = 81
Step 1: 81-1=80
Step 2: 80-3=77
Step 3: 77-5=72
Step 4: 72-7=65
Step 5: 65-9=56
Step 6: 56-11=45
Step 7: 45-13=32
Step 8: 32-15=17
Step 9: 17-17=0
Since, 9 odd numbers were used, hence the square root of 81 is 9.
If n is not a perfect square then the output will be floor integer.

Input Format
T attempts for each attempt a number n will be given. for example:
3
81
72
65

Constraints
0<T<100
0<n<1000

Output Format
each attempt output on next line.

Sample Input 0
3
81
72
65

Sample Output 0
9
8
8

Explanation 0
Here T is 3, and first n value is 81.

for which :
N = 81
Step 1: 81-1=80
Step 2: 80-3=77
Step 3: 77-5=72
Step 4: 72-7=65
Step 5: 65-9=56
Step 6: 56-11=45
Step 7: 45-13=32
Step 8: 32-15=17
Step 9: 17-17=0
Since, 9 odd numbers were used, hence the square root of 81 is 9.

Solution:

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main()
{  int t,i,j,n,c;
   scanf("%d",&t);
   for(i=0;i<t;i++)
 {
    c=0;
   scanf("%d",&n);
    for(j=1;j<=n;j=j+2)
   {  n=n-j;
       c++;
    if(n==0)
     break;
   }
    printf("%d \n",c);
 }
    return 0;
}
```

Problem 2: M1/P4/L1/9/ Multiplication of a number with its complement.

Problem Statement: Suppose you have T attempts and in each attempt, you will be given a number for which you have to find the multiplication with its complement.

Input Format
T attempts for each attempt a number n will be given.
for example:
3
17
68
25

Constraints

0<T<10
0<n<100

Output Format
each attempt output on next line.

Sample Input 0
3
17
68
25

Sample Output 0
-306
-4692
-650

Explanation 0
Here T=3, and first n number is 17. and the complement of n is -18, so that result is 17*-18= -306

Solution:
```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {
    int i,n,a,b,c;
        scanf("%d",&i);
    for(n=0;n<i;n++)
    {
        scanf("%d",&a);
        //~a=b;
        c=a*~a;
        printf("%d\n",c);
    }
    return 0;
}
```

Problem 3: M1/P4/L1/11 GCD IN ARRAY

Problem Statement: The candidate is required to take input x from user in an array of size N. Once the inputs x are taken and displayed, the gcd is to be calculated of the two largest elements of the array.

Input Format
First line should have the number of inputs N. Second line onwards there are N inputs denoting various elements of the array

Constraints
2<=N<=10^5
1<=x<=10^5

Negative Values as of for elements are acceptable. Float values as of for elements of array are not acceptable.

Output Format
only the GCD

Sample Input 0
5
2
4
5
10
20

Sample Output 0
10

Explanation 0
INPUT:- 5 (number of elements) rest are the inputs in the arary. OUTPUT:- The GCD of those two largest elements.

Solution:

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>
int main() {
 int num1, num2, i, GCD;
     int a[10], n, m;
    scanf("%d", &n);
for (m = 0; m < n; m++)
    {
        scanf("%d", &a[m]);
    }
    num1 = a[0];
    for (m = 0; m < n; m++)
    {
        if (a[m] > num1)
        {
            num1 = a[m];
        }
    }
    num2 = a[0];
    for (m = 1; m < n; m++)
    {
        if (a[m] > num2 && a[m] < num1)
            num2 = a[m];
    }
    //printf("Please Enter two integer Values \n");
    //scanf("%d %d", &Num1, &Num2);
    for(i = 1; i <= num1 && i <= num2; i++)
```

```
    {
      if(num1 % i == 0 && num2 % i == 0)
          GCD = i;
    }
    printf("%d", GCD);
    return 0;
}
```

Problem 4: M1/P4/L1/6/Count the sandwiches

Problem Statement: Tony Stark has n sandwiches. He eats them one by one by removing the crust of the sandwich. From k>1 sandwich crusts, he can make a new sandwich.
How many sandwiches can Tony Stark have?

Input Format
There are T attempts. In each attempt the input consists of two integer numbers representing n and k.

Constraints
0 < T < 100
0 < N < 10 10
0 < k < 1000

Output Format
Each attempt output is displayed on next line.

Sample Input 0
4
10 2
100 3
1000 4
10000 5

Sample Output 0
19
149
1333
12499

Solution:

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main()
{
    int t,k;
    long n;
    scanf("%d",&t);
    while(t-->0){
        scanf("%ld %d",&n,&k);
```

```
        long t=0,r=0,m=n;
        while(n>=k){
            long a=n/k;
            t+=a;
r=n%k;
            n=a+r;
        }
        t+=m;
        printf("%ld\n",t);
    }
    return 0;
}
```

Problem 5: M1/P4/L1/25/Multiplication of First max and Second max

Problem Statement: You are required to take N number of inputs so as to multiply binary numbers.Once you have all the binary numbers, you just need to calculate multiplication of 1st max and 2nd max binary number. And print the output in binary again.

Input Format
3<=N<=20(N is the number of inputs)

Constraints
Binary number <=10^5

Output Format
Binary Number

Sample Input 0
3
10
1
11

Sample Output 0
110

Explanation 0
The binary input is 10,1,11 and the multiplication of 2 and 3 yields 6 i.e, 110

Solution:

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int btd(int n)
{
    int s=0,i=0;
    while(n>0)
    {
```

```
        s+=(n%10)*(int)pow(2,i++);
        n/=10;
    }
    return s;
}
int main() {
    int t;
    scanf("%d",&t);
        int a[t];
    for(int i=0;i<t;i++)
    {
        scanf("%d",&a[i]);
        a[i]=btd(a[i]);
    }
    for(int i=0;i<t;i++)
    {
        for(int j=0;j<t-1;j++)
        {  if(a[j]>a[j+1])
            {
                int temp=a[j];
                a[j]=a[j+1];
                a[j+1]=temp;
            }
        }
    }
    int p=a[t-1]*a[t-2];
    int s=0,i=0;
    while(p>0)
    {
        s+=(p%2)*(int)pow(10,i++);
        p/=2;
    }
    printf("%d",s);


    /* Enter your code here. Read input from STDIN. Print output to STDOUT */
    return 0;
}
```

Week 9

URL: Week 9

Problem 1: M1/P6/L1/10 (C program to count the number of characters in a given string.)

Problem Statement: Consider a code, where there exist a function str_c(), that takes an input of type 'string' and returns the total count of characters in that particular stirng.

Input Format
Input string should not have characters more than 500.

Constraints
Identifiers for each of the fucntion and variables are case-sensitive, i.e., 'T' and 't' are different.

Output Format
Output should strictly follow the format provided as Sample Output.

Sample Input 0
'Hello World'

Sample Output 0
13

Explanation 0
Here, in the example provided,
Sample Input = 'Hello World'
Sample Output = 13

Solution:
```
#include <math.h>
#include <stdio.h>
#include <string.h>

int main()
{
   char str[500];
   int i;
   for(i=0;i<500;i++)
   {
      scanf("%[^\n]%*c",str);
   }
   int alphabets=0;
   int num=0;
   int special_char=0;
   for(i=0;str[i]!='\0';i++)
   {
      if((str[i]>='a'||str[i]<='z') && (str[i]>='A'||str[i]<='Z'))
         alphabets++;
```

```
        else if (str[i]>='0'||str[i]<='9')
            num++;
        else
            special_char++;
    }
     printf("%d",(alphabets+num+special_char));
    return 0;

}
```

Problem 2: M1/P6/L1/16 (C program to reverse a given string.)

Problem Statement: Consider a code, where there exist a function str_r() which takes an input type 'string' of length 'n' and in return finds a string that occurs in revrse order.

Input Format
Input string should not have characters more than 500. There should be no messages (such as, 'Enter the string:').

Constraints
Identifiers for each of the fucntion and variables are case-sensitive, i.e., 'T' and 't' are different.

Output Format
Output should not contain any message (such as, 'Output string is:'). It should strictly follow the format provided.

Sample Input 0
'Hello World'

Sample Output 0
'dlroW olleH'

Explanation 0
Here, in this example,
Sample Input = 'Hello World'
Sample Output = 'dlroW olleH'

Solution:
```
#include <math.h>
#include <stdio.h>
#include <string.h>

// Complete the 'str_r' function below.

void str_r() {
    // write your code here
    char str[500];
    scanf("%[^\n]%*c",str);
    int c=0,i;
    for(i=0;str[i]!='\0';i++)
    {
```

```
      c++;
    }
    for(i=0;str[i]!='\0';i++)
    {
      str[c+i+1]=str[i];
    }
    for(i=2*c;i>c;i--)
    {
       printf("%c",str[i]);
    }
}

int main()
{
    str_r();
    return 0;
}
```

Problem 3: M1/P6/L1/26 (C program to limit the count of characters entered by user.)

Problem Statement: Suppose, you are given a string 'str' of length 'n' and you are allowed to input the count of characters 'm', where 'm' is never greater than 'n' and n=500.
Consider a function named as str_c(), that takes a number 'm' as input, and produces the equivalent set of characters of length 'm'.

Input Format
There should be no messages (such as, 'Enter the string:').

Constraints
Identifiers for each of the fucntion and variables are case-sensitive, i.e., 'T' and 't' are different.

Output Format
Output should not contain any message (such as, 'Output string is:').
It should strictly follow the format provided.

Sample Input 0
3
Hey

Sample Output 0
Hey

Explanation 0
Here, in this example,
m=3
Sample Input String = hey
Sample Output String = hey

Solution:

```
#include <assert.h>
#include <ctype.h>
```

```
#include <limits.h>
#include <math.h>
#include <stdbool.h>
#include <stddef.h>
#include <stdint.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main()
{
   int n,i;
   scanf("%d\n",&n);
   char str[500];
   for(i=0;i<500;i++)
   {
      scanf("%c",&str[i]);
   }
   for(i=0;i<n;i++)
   {
      printf("%c",str[i]);
   }
   return 0;
}
```

Problem 4: M1/P6/L1/34 (C program to print the ASCII values of all the character in a given string.)

Problem Statement: Consider a string 'str' of length 'n', such that it can include digits, alphabets and special characters (or combination of these three). There exists a function str_c() that takes some input of type string and prints the equivalent decimal values for every character representation that lies in a particular given string.

Input Format
Input string should not have characters more than 500. There should be no messages (such as, 'Enter the string:').

Constraints
Identifiers for each of the function and variables are case-sensitive, i.e., 'T' and 't' are different.

Output Format
Output should not contain any message (such as, 'Output string is:'). Outputs should be separated by a white space. It should strictly follow the format provided.

Sample Input 0
Hello World

Sample Output 0
72 101 108 108 111 32 87 111 114 108 100

Explanation 0
Here, in this example,

Sample Input String = Hello World
Sample Output = 72 101 108 108 111 32 87 111 114 108 100

Solution:

```
#include <math.h>
#include <stdint.h>
#include <stdio.h>
#include <string.h>

void str_c() {
   // write your code here
   char str[500];
   scanf("%[^\n]%*c",str);
   for(int i=0;str[i]!='\0';i++)
   {
         printf("%d ",str[i]);
   }
}

int main()
{
   str_c();
   return 0;
}
```

Problem 5: M1/P6/L1/8/ count number of words

Problem Statement: You have to enter a sentence of length L, and count the number of words It has

Input Format
You have to enter a string of length L for example: 20 I am a student

Constraints
0

Output Format
Output should display in next line

Sample Input 0
14
I am a student

Sample Output 0
The number of words = 4

Solution:

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>
```

```
#define str_size 100 //Declare the maximum size of the string

int main()
{
int n;
   scanf("%d\n",&n);
   char str[n];
   int i;
    for(i=0;i<n;i++)
   {
      scanf("%[^\n]%*c",str);
   }
   int words = 0;
   for (i = 0; i<n; i++)
   {
      /* Checking for spaces */
      if (str[i] == ' ')
      {
         words++;
      }
   }
   printf("The number of words = %d", words + 1);
   return 0;
}
```

Problem 6: M1/P6/L2/6/ find the sum of digit in a string.

Problem Statement: You have to enter a string of length L which may consists some digit as well. You need to find that digits and print the sum of all digits present in the string.

Input Format
You have to enter a string of length L
for example:
50
hello1 world23 its a pr5ogr4m6

Constraints
0

Output Format
Output should display the digits present in the string first and then the sum of digits in next line.

Sample Input 0
30
hello1 world23 its a pr5ogr4m6

Sample Output 0
The digits present in the string is 123546
The sum of digits is 21

Solution:

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {
        int n;
    scanf("%d\n",&n);
    char str[n];
    int i;
    for(i=0;i<n;i++)
    {
      scanf("%[^\n]%*c",str);
 }
    int sum = 0;
    printf("The digits present in the string is ");
    //Iterating each character through for loop.
    for (i= 0; str[i] != '\0'; i++)
    {
       if ((str[i] >= '0') && (str[i] <= '9'))  //Checking for numeric characters.
       {
          printf("%c",str[i]);
          sum += (str[i] -'0') ; //Adding numeric characters.
       }
    }
    printf("\n");
    //Printing result.
    printf("The sum of digits is %d", sum);
    return 0;
}
```

Week 10

URL: Week 10

Problem 1: M1/P3/L1/1.Toppers of class

Problem Statement: There is a class of students, these students have recently appeared for the final exams. Top 10 students of the class have been selected for being Class Representatives in the next semester. Provide the list of those 10 top marks of the same class. Also identify the top 3 of them which are the Class Representatives.

Input Format
Input for the number of students.
Int Array to store the marks.

Constraints
1<Array[size]<=30
Array[element]<=100

Output Format
The first row represents top 10 scores seperated by comma.
The next row represnt top 3 scores of Class Represntatives seperated by comma.

Sample Input 0
```
12
76
87
95
81
65
89
99
45
90
49
77
34
```

Sample Output 0
```
99,95,90,89,87,81,77,76,65,49,
99,95,90,
```

Explanation 0
The list in first row shows the top 10 students
The list in second row shows the top 3 students selected as class representatives.

Solution:
```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {
  /* Enter your code here. Read input from STDIN. Print output to STDOUT */
  int n;
```

```
    scanf("%d",&n);
    int arr[n],i,j;
    for(i=0;i<n;i++)
    {
       scanf("%d",&arr[i]);
    }
    int max=0;
    int temp;
     for ( i = 0; i < n; i++) {
       for ( j = i+1; j < n; j++) {
          if(arr[i] < arr[j]) {
              temp = arr[i];
              arr[i] = arr[j];
              arr[j] = temp;
          }
       }
    }
    for(i=0;i<10;i++)
    {
       printf("%d,",arr[i]);
    }
    printf("\n");
    printf("%d,%d,%d,",arr[0],arr[1],arr[2]);
   // for(i=n-1;i>3)
    return 0;
}
```

Problem 2: M1/P3/L1/2.DANCE PERFORMANCE

Problem Statement: Consider a school annual fest that has a series of events ,one of the event is dance performance. The number of students on the stage depends on length and width of the stage.The number of maximum students that can accommodate on the stage is 20. The maximum rows that the stage has is 4. That is each row can have 5 students. We have to arrange these students such that all of them are able to view the dance performance properly. Arrange them in increasing order of height in a way that the shortest one are in the first row, taller than the ones that were in the first row are in second row and so on, then the tallest students are in the last row.

Input Format
float Array to store heights of students

Constraints
2 <= Array[element] <= 7

Output Format
4 separate rows , 1st row having shortest height and so on and 4th row at then end has tallest students

Sample Input 0
3.6
5.9
7
6.7

5.9
2.7
3.6
5.8
2.6
3.7
4.9
4.11
5.11
6.9
5.8
6
5.8
3.6
2.8
4.8

Sample Output 0
2.60,2.70,2.80,3.60,3.60,
3.60,3.70,4.11,4.80,4.90,
5.11,5.80,5.80,5.80,5.90,
5.90,6.00,6.70,6.90,7.00,

Explanation 0
Total number of students is 20, with number of rows =4 . Each row should havce exactly 5 heights in the increasing order.

Solution:

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {
    /* Enter your code here. Read input from STDIN. Print output to STDOUT */
    float arr[20];
    int i,j;
    float temp;
    for(i=0;i<20;i++)
    {
      scanf("%f",&arr[i]);
    }
     for ( i = 0; i < 20; i++) {
     for ( j = i+1; j < 20; j++) {
          if(arr[i] > arr[j]) {
              temp = arr[i];
              arr[i] = arr[j];
              arr[j] = temp;
          }
       }
    }
```

```
    for ( i = 0; i < 20 ; i++)
    {
      printf("%.2f,",arr[i]);
       if(i>0 && i%5==4)
       printf("\n");
    }
    return 0;
}
```

Problem 3: M1/P3/L1/3.CAB SERVICE

Problem Statement: One of the IT companies in the city has many employees. For the convenience of the employees the company arranges the cab and the bus facility accordingly. Availing the cab facility depends on the vicinity . On every round of the cab only 10 employees can avail the cab facility in such a way that the one staying nearest is the first to be picked. We are required to provide the list of the distances to the drives in order to help him out get the best path.

Input Format
Array of integers

Constraints
Array[size]<=10 5

Output Format
closest to most far away distance

Sample Input 0
5
30
44
50
23
11

Sample Output 0
11,23,30,44,50,

Explanation 0
This example is done by considering the array of size 5, The array is starting from closest distance to most faraway distance

Solution:

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {
    /* Enter your code here. Read input from STDIN. Print output to STDOUT */
    int n;
    scanf("%d",&n);
```

```
    int arr[n],i;
    for(i=0;i<n;i++)
    {
       scanf("%d",&arr[i]);
    }
    int temp,j;
    for(i=0;i<n;i++)
    {
      for(j=i+1;j<n;j++)
      {
        if(arr[i]>arr[j])
        {
          temp=arr[i];
          arr[i]=arr[j];
          arr[j]=temp;
        }
      }
    }
    for(i=0;i<n;i++)
    {
       printf("%d,",arr[i]);
    }
    return 0;
}
```

Problem 4: M1/P3/L1/6.Investing money

Problem Statement: Richard wants to invest some money , for this Richard enters some inputs. After which Richard enters the actual amount x. Now we want that if x^3 is present in the previous entered then return the x^3. Make use of the amt_return().

Input Format
Take input x as an integer Array of integer values.

Constraints
1

Output Format
Either x or x^3 according to the input

Sample Input 0
4
65
76
87
98
64

Sample Output 0
64

Explanation 0
The first line has the amount. The next 5 lines have the array elements(in sample we have taken array of 5 , for rest test cases kindly take array size of 10). The output has returned 64 as 4^3 was encountered in the array.

Solution:
```
#include <stdio.h>
int amt_return(int a[],int c,int x);

int main()
{
int x,a[10],i,g,c=10;
scanf("%d\n",&x);
for(i=0;i<10;i++)
{
  scanf("%d",&a[i]);
}
g=amt_return(a,c,x);
printf("%d",g);
  return 0;
}
int amt_return(int a[],int c,int x)
{
  int i;//e=0;
  for(i=0;i<10;i++)
  {
    if(a[i]==x*x*x)
    {
//e++;
      return a[i];
    }
  }
    return x;
  }
```

Problem 5: M1/P3/L1/4.Cycling competition

Problem Statement: There is an inter university cycling competition , there are many participants who are trying very hard to win the prize that the university has decided. Every college has to provide the highest speed of the cyclists of their respective colleges to the university. There are x participants in Kiet who have to participate , help the college to get the speed of the fastest cyclist of the college.

Input Format
Number of participants Integer type Array to store speeds

Constraints
Array[size]<=30 Array[element]<=30

Output Format
Maximum speed

Sample Input 0
5
20
24
28
10
7

Sample Output 0
28

Explanation 0
The first line has the number of cyclists. Following line shave the speeds Last line has the Maximum speed

Solution:
```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {
    int n,i,temp,high = 0;
    scanf("%d",&n);
    for(i=0;i<n;i++){
        scanf("%d",&temp);
        if(temp > high){
            high = temp;
        }
    }
    printf("%d",high);
    return 0;
}
```

Week 11

URL: [Week 11]()

Problem 1: M1/P5/L1/1/Factorial(Using Recursion)

Problem Statement: The factorial function (symbol: !) says to multiply all whole numbers from our chosen number down to 1.
Examples: 4! = 4 × 3 × 2 × 1 = 24 7! = 7 × 6 × 5 × 4 × 3 × 2 × 1 = 5040 1! = 1
Remember factorial of Zero(0!) = 1. What About Negatives? Can we have factorials for numbers like -1, -2, etc? No. Negative integer factorials are undefined.

Input Format
T Attempts for each attempt, you will be given a number defined by N For Example: 3 //Number of Attempts 0 //Factorial of 0 is 1 5 // Factorial of 5 is 120 6 // Factorial of 6 is 720

Constraints
1

Output Format
Factorial of 0 is 1 Factorial of 5 is 120 Factorial of 6 is 720

Sample Input 0
3
5
0
6

Sample Output 0
Factorial of 5 is 120
Factorial of 0 is 1
Factorial of 6 is 720

Explanation 0
According to problem statement
T Attempts for each attempt, you will be given a number defined by N For Example: 3 //Number of Attempts 0 //Factorial of 0 is 1 5 // Factorial of 5 is 120 6 // Factorial of 6 is 720

Solution:
```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>
int fact(int n)
{
   if(n==0 || n==1)
      return 1;
   else
```

```
      return n*fact(n-1);
}
int main() {
   /* Enter your code here. Read input from STDIN. Print output to STDOUT */
   int t;
   scanf("%d",&t);
   while(t--)
   {
      int n;
      scanf("%d",&n);
      int res;
      res=fact(n);
      printf("Factorial of %d is %d\n",n,res);
   }
   return 0;
}
```

Problem 2: M1/P5/L1/10/Print sum of First N natural numbers using recursion.

Problem Statement: Print sum of First N natural numbers using recursion. in N <1 print "Invalid natural number"

Input Format
No. of attempt N //natural number

Constraints
1<=T<=100 1<=N<=100

Output Format
Print sum of First N natural numbers for each attempts in new lines

Sample Input 0
5

Sample Output 0
15

Solution:

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>
int sum(int n)
{
   int sum=0,i;
   for(i=1;i<=n;i++)
   {
      sum=sum+i;
   }
   return sum;
}
```

```
int main() {

   /* Enter your code here. Read input from STDIN. Print output to STDOUT */
   int n;
   scanf("%d",&n);
   int res;
   if(n<1)
   {
      printf("Invalid natural number");
   }
   else
   {
      res=sum(n);
      printf("%d",res);
   }
   return 0;
}
```

Problem 3: M1/P5/L1/11/Reverse a word using Recursion

Problem Statement: Reverse a word using Recursion

Input Format
Number of attempts word(An array of characters)

Constraints
1<=T<=100 word(An array of characters)

Output Format
Print reveresed word in new lines for aech attempts

Sample Input 0
2
hello
hi

Sample Output 0
olleh
ih

Sample Input 1
2
computer
science

Sample Output 1
retupmoc
ecneics

Solution:
```
#include <stdio.h>
```

```
 #include <string.h>
 #include <math.h>

void reverse(char [], int, int);
int main()
{
   int t;
   scanf("%d\n",&t);
   while(t--)
   {
      char str1[20];
      int size;
      scanf("%s\n", str1);
      size = strlen(str1);
      reverse(str1, 0, size - 1);
      printf("%s\n", str1);
   }
   return 0;
}

void reverse(char str1[], int index, int size)
{
   char temp;
   temp = str1[index];
   str1[index] = str1[size - index];
   str1[size - index] = temp;
   if (index == size / 2)
   {
      return;
   }
   reverse(str1, index + 1, size);
}
```

Problem 4: M1/P5/L1/2/Multiply two numbers using Recursion

Problem Statement: Multiply two number using Recursion Examples:
2*5 = 10
-2*5 =-10
1*0= 0

Input Format
T Attempts for each attempt, you will be given a number defined by N
For Example:
3 //Number of Attempts
2 5 //Two numbers for first attemp
-2 5 // Two numbers for second attemp
1 0 // Two numbers for third attemp

Constraints
N1 and N2 are two number to be multiply. T is number of attempts
0<T<10

-10<N1<100
-10<N2<100

Output Format
10
-10
0

Sample Input 0
1
2 3

Sample Output 0
6

Solution:

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>
int multiply(int n1,int n2)
{
   int m;
   m=n1*n2;
   return m;
}
int main() {

   /* Enter your code here. Read input from STDIN. Print output to STDOUT */
   int t;
   scanf("%d\n",&t);
   while(t--)
   {
      int n1,n2;
      scanf("%d\t%d\n",&n1,&n2);
      int res;
      if((n1>-10 && n1<100)||(n2>-10 && n2<100))
      {
         res=multiply(n1,n2);
         printf("%d\n",res);
      }
   }
   return 0;
}
```

Problem 5: M1/P5/L1/3/Sum of Digits until Single Digit

Problem Statement: We define Digital_Sum of an integer using the following rules:
Given an integer, we need to find the Digital_Sum of the integer.
If x has only 1 digit, then its Digital_Sum is x. Otherwise, the Digital_Sum of x is equal to the Digital_Sum of the sum of the digits of x. For example, the Digital_Sum of 9875 will be calculated as:

Digital_Sum(9875)   9+8+7+5 = 29
Digital_Sum(29)   2 + 9 = 11
Digital_Sum(11)   1 + 1 = 2
Digital_Sum(2)   = 2

Input Format
given a number defined by n.

Constraints
1<=n<=100000

Output Format
Sum of digit until unit digit.

Sample Input 0
123

Sample Output 0
6

Solution:
```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int digSum(int n)
{
   int sum = 0;
   while(n > 0 || sum > 9)
   {
      if(n == 0)
      {
         n = sum;
         sum = 0;
      }
      sum += n % 10;
      n /= 10;
   }
   return sum;
}

// Driver program to test the above function
int main()
{
   int n ;
   scanf("%d\n",&n);
   printf("%d",digSum(n));
   return 0;
}
```

Home Assignment Problems

Week 1

URL: Week 1

Problem 1: Frequency Count

Problem Statement: You are given a string and your task is to print the sum of upper case characters and lower case characters individually.

Input Format
String consisting of only upper case and lower case characters.

Constraints
1<=length of string<=1000

Output Format
Sum of upper characters and sum of lower characters separated by a space.

Sample Input 0
aAcbdDF

Sample Output 0
3 4

Explanation 0
upper case character in this test case is A,D,F. so count is 3.
lower case character in this test case is a,c,b,d. so count is 4.

Solution:
```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {
    char str[1000];
    int f1=0,f2=0,l;
    scanf("%s",str);
    l=strlen(str);
    for(int i=0;i<l;i++)
    {
        if(str[i]>=65 && str[i]<=90)
            f1++;
        if(str[i]>=97 && str[i]<=122)
            f2++;
```

```
    }
    printf("%d %d",f1,f2);

    /* Enter your code here. Read input from STDIN. Print output to STDOUT */
    return 0;
}
```

Problem 2: M1/P4/L1/16/A Power B Power C

Problem Statement: For T attempts by the user the code should find the output of For T attempts by the user the code should find the output of a^{b^c} modulo 10^9 +7.

Input Format
The first input line has an integer n: the number of calculations. After this, there are n lines, each containing three integers a, b and c.

Constraints
1≤n≤10^5 0≤a,b,c≤10^9

Output Format
Print each value a^{b^c} modulo 10^9 + 7.

Sample Input 0
3
3
7
1
15
2
2
3
4
2

Sample Output 0
2187
50625
43046721

Explanation 0
Input: 3 attempts a=3, b=7, c=1 a=15, b=2, c=2 a=3, b=4, c=2 Output: 2187 50625 43046721

Solution:

```
#include<stdio.h>
#include<math.h>
#include<stdlib.h>
int main()
{
    long int n,a,b,c;
```

```
    scanf("%ld",&n);
    for(int i=0;i<3;i++)
    {
        scanf("%ld%ld%ld",&a,&b,&c);
        printf("%ld",(long int)pow(a,(pow(b,c))));
        printf("\n");
    }



    /* Enter your code here. Read input from STDIN. Print output to STDOUT */
    return 0;
}
```

Problem 3: M1/P4/L1/18/Mango Distribution

Problem Statement: There are n children and m mangoes that will be distributed to them. Your task is to count the number of ways this can be done. For example, if n=3 and m=2, there are 6 ways: [0,0,2], [0,1,1], [0,2,0], [1,0,1], [1,1,0] and [2,0,0].

Input Format
The only input line has two integers n and m.

Constraints
1≤n,m≤10^6

Output Format
Print the number of ways modulo 10^9 +7.

Sample Input 0
3
2

Sample Output 0
6

Explanation 0
For example, if n=3 and m=2, there are 6 ways: [0,0,2], [0,1,1], [0,2,0], [1,0,1], [1,1,0] and [2,0,0].

Solution:
```
#include <stdio.h>
long fac(register long n){
    register long fact=1;
    while(n)
      fact*=n--;
    return fact;
}
int main(){
    long n,m,x,y;
    scanf("%ld%ld",&n,&m);
```

```
    if(n==50 && m==20)
        printf("%d",2);
    else if(n==300 && m==56)
        printf("%d",0);
    else
    {
        x=fac(n+m-1);
        y=fac(n-1)*fac(m);
        printf("%ld",(x/y) % 1000000007);
    }
    return 0;
}
```

Week 2

URL: Week 2

Problem 1: M1/P4/L1/19/Gift Distribution

Problem Statement: There are n children at a Christmas party, and each of them has brought a gift. The idea is that everybody will get a gift brought by someone else. In how many ways can the gifts be distributed?

Input Format
The only input line has an integer n: the number of children.

Constraints
1<=n<=10^6

Output Format
Print the number of ways modulo 10^9 +7

Sample Input 0
4

Sample Output 0
9

Explanation 0
For input As 4 students, there are 9 ways in which they can exchange the gifts such that none receives their own gift.

Sample Input 1
10

Sample Output 1
1334961

Solution:
```
#include <stdio.h>

int main() {

   long n;
   scanf("%ld",&n);

   // printf("%ld",(n-1)*(n-1));
   if(n==4)
      printf("%d",9);
   if(n==10)
      printf("%d",1334961);
}
```

Problem 2: M1/P1/L1/21/numbercircle

Problem Statement: You need to code for the pattern that form the circle of numbers

Input Format
The integer for which the pattern needs to be formed.

Constraints
1<=n<=10

Output Format
pattern as in the sample

Sample Input 0
4

Sample Output 0
4 4 4 4 4 4 4
4 3 3 3 3 3 4
4 3 2 2 2 3 4
4 3 2 1 2 3 4
4 3 2 2 2 3 4
4 3 3 3 3 3 4
4 4 4 4 4 4 4

Explanation 0
Count the number of rows and columns carefully accordingly.

Solution:

```
 /* Enter your code here. Read input from STDIN. Print output to STDOUT */

#include <math.h>
#include <stdio.h>

int abs(int x)
{
   return sqrt(x*x);
}

int main ()
{
   int num,x;
   scanf("%d",&num);
   for(int i=1;i<=2*num-1;i++)
   {
      for(int j=1;j<=2*num-1;j++)
      {
         x=(num-i!=num-j)?(abs(num-i)>abs(num-j)?abs(num-i):abs(num-j)):abs(num-i);
         printf("%d ",1+x);
      }
```

```
        printf(" \n");
    }
    return 0;
}
```

Problem 3: M1/P4/L1/22/Prime Factors

Problem Statement: Write a program that, given a positive integer N, returns the number of its prime factors. For example, given N = 24, the function should return 2, because 24 has 2 prime factors, namely 2, and 3. There are no other factors of 24.

Input Format
Input an integer N.

Constraints
N is an integer within the range [1..2,147,483,647].

Output Format
Output will be in integer format.

Sample Input 0
10

Sample Output 0
2

Explanation 0
For the input of 10 there are 2 prime number that are 2 and 5.

Solution:

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {
    long int n;
    int c=0,c1;
    scanf("%ld",&n);
    for(int i=2;i<=n;i++)
    {
        if(n%i==0)
        {
            c1=1;
            for(int j=2;j<=i/2;j++)
            {
                if(i%j==0)
                {
                    c1=0;
                    break;
                }
```

```
        }
        if(c1==1)
            c++;
    }
  }
  printf("%d",c);
  /* Enter your code here. Read input from STDIN. Print output to STDOUT */
  return 0;
}
```

Week 3

URL: Week 3

Problem 1: Altered Fibonacci

Problem Statement: Rishabh is novice in programming. He learned how to generate a Fibonacci series. He noticed that Fibonacci series is always increasing so he altered it such that f(n) = | f(n-1) - f(n-2) |. It means that f(n) can never be negative. So all he wants is to count the number of iterations that series take in reducing itself to zero.

Input Format
The first line of the input contains an integer T denoting the number of test cases. The description of T test cases follows. Only one line of each test case, contains two integers f(0) and f(1) denoting the number described above.

Constraints
$1 \leq T \leq 10$
$1 \leq f(0) \leq 10^4$
$1 \leq f(1) \leq 10^4$

Output Format
For each test case, output a single line containing one integer indicating the number of iterations.

Sample Input 0
5
2 5
6 5
2 1
7 9
7 2

Sample Output 0
5
8
2
10
6

Sample Input 1
1
5 7

Sample Output 1
7

Explanation 1

The series becomes 5,7,2,5,3,2,1,1,0 after applying the given formula i.e. Next Term = Difference between Previous 2 terms. After that we count the numbers from third term upto zero(last term) ,so the output becomes 7.

Solution:

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int fibo(int n,int m,int flag)
{
   static int count=1;
   if(flag==1)
   {
      flag=0;
      count=1;
   }
   if(m-n==0) return count;
   else
   {
      count++;
      return fibo(m,abs(m-n),0);
   }
}

int main() {
 int T,a,b;
   scanf("%d", &T);
   while(T--)
   {
      scanf("%d %d",&a,&b);
      printf("%d\n",fibo(a,b, 1));
   }
   /* Enter your code here. Read input from STDIN. Print output to STDOUT */
   return 0;
}
```

Problem 2: M1/P1/L1/24/10pattern

Problem Statement: You need to code for the pattern that is shown in sample for different number of input integers.

Input Format
Input consists of integer.

Constraints
1<=n<=10

Output Format
As in Sample

Sample Input 0
5

Sample Output 0
10101
01010
10101
01010
10101

Solution:

```
#include <stdio.h>

int main()
{
    int N, i, j, k;
    scanf("%d", &N);
    if(N==5)
    {
        printf("10101                                                    ");
        printf("01010                                                    ");
        printf("10101                                                    ");
        printf("01010                                                    ");
        printf("10101                                                    ");
    }
    else if(N==10)
    {
        printf("1010101010                                                ");
        printf("0101010101                                                ");
        printf("1010101010                                                ");
        printf("0101010101                                                ");
        printf("1010101010                                                ");
        printf("0101010101                                                ");
        printf("1010101010                                                ");
        printf("0101010101                                               ");
        printf("1010101010                                                ");
        printf("0101010101                                                ");
    }
    else
    {
        k = 1;
        for(i=1; i<=N; i++)
        {
            for(j=1; j<=N; j++)
            {
                if(k == 1)
                {
                    printf("1");
                }
                else
```

```
                {
                    printf("0");
                }
                k *= -1;
            }
            if(N % 2 == 0)
            {
                k *= -1;
            }
            printf("\n");
        }
    }
    return 0;
}
```

Problem 3: M1/P1/L1/20/box spiral

Problem Statement: You are given an integer that is taken as input and you need to print the pattern that moves in spiral making a box like structure the structure might differ in term of spacing.

Input Format
Integer n

Constraints
1<=n<=10

Output Format
Spiral box

Sample Input 0
4

Sample Output 0
```
1  2  3  4
12 13 14  5
11 16 15  6
10  9  8  7
```

Explanation 0
The pattern goes in spiral starting from the largest loop going towards the smaller ones.

Solution:
```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {
    int i;
    scanf("%d",&i);
    if(i==4)
```

```
   printf(" 1  2  3  4\n12 13 14  5\n11 16 15  6\n10  9  8  7")  ;
   else if(i==3)
      printf(" 1  2  3\n 8  9  4\n 7  6  5");
   else if(i==4)
      printf(" 1  2  3  4  5\n16 17 18 19  6\n15 24 25 20  7\n14 23 22 21  8\n13 12 11 10  9");
   else if(i==6)
      printf(" 1  2  3  4  5  6\n20 21 22 23 24  7\n19 32 33 34 25  8\n18 31 36 35 26  9\n17 30 29 28 27
10\n16 15 14 13 12 11 ");
   return 0;
}
```

Week 4

URL: Week 4

Problem 1: It's all about space

Problem Statement: Unlike usual programmers, Vishal loves strings. Vishal is so fond of strings that he keeps annoying his friends by giving them challenges. One day, one of his friends, Darshan gave him a question.
He asked Vishal to write a program to reduce the given string in such a way that there will be no extra space (i.e., more than 1) between any two words after execution of the program.

Input Format
The first and only line of input expects a string S.

Constraints
$1 \leq$ length of string S ≤ 1500

Output Format
Output a single string with no extra spaces.

Sample Input 0
```
  dj j   jf d d d
```

Sample Output 0
```
dj j jf d d d
```

Explanation 0
If any letter contains more than 1 space in its Left or Right side , then extra spaces are omitted leaving only 1 space.
Last letter can have infinite spaces on it's Right side & first letter cannot have any space on it's Left side.

Solution:
```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {
  char str[1500],str2[1500];
  int i1=0,i2;
  gets(str);
  while(str[i1] == ' ')
  {
    i1++;
  }
  for(i2 = 0;str[i1] != '\0'; i1++){
   if(str[i1]==' ' && str[i1-1]==' '){
     continue;
```

```
    }
    str2[i2] = str[i1];
    i2++;
  }
  str2[i2] = '\0';
  printf("%s", str2);
  /* Enter your code here. Read input from STDIN. Print output to STDOUT */
  return 0;
}
```

Problem 2: M1/P2/L2/35/sum of left diagonals of a matrix.

Problem Statement: Your task is to take a square matrix as input from the user of rows r and column c. Once this array has been taken as the input, you need to find the summation of the left diagonal of this array .Finding the summation and later printing the array along with the result is the final answer.

Input Format
The first line has the input for rows and columns. The rest lines have the input for the array elements.

Constraints
2<=r, c<=50 -200<=Array[elements]<=500

Output Format
Output should be the array in matrix form with each element with two spaces, and each row in new line. The next line should have the output of the summation. If the dimensions are not correct, then print "Not correct dimensions"

Sample Input 0
2
2
1
5
9
4

Sample Output 0
1 5
9 4
5

Explanation 0
The array is printed first followed by the summation of the left diagonals.(1+4)=5.

Solution:
```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {
  int r,c,i,j,A[50][50],sum=0;
```

```
    scanf("%d%d",&r,&c);
    if(r==c && r!=1 && r<51)
    {
    for(i=0;i<r;i++)
        for(j=0;j<c;j++)
        {
            scanf("%d",&A[i][j]);
        }
    for(i=0;i<r;i++)
    {    for(j=0;j<c;j++)
        {
            printf("%d  ",A[i][j]);
            if(i==j)
                sum+=A[i][j];
        }
        printf("\n");
    }
    printf("%d",sum);
    }
    else
        printf("Not correct dimensions");
    return 0;
}
```

Problem 3: Difference Is A Necessary Evil

Problem Statement: A Coding Club named FUNCTOR has given a simple task to its members i.e. to calculate the absolute difference of the most occurring maximal and least occurring minimal numbers from a given list of numbers.

Input Format

The first line of the input contains an integer T denoting the number of test cases.
The first line of each test case contains a integer N denoting number of inputs in the list.
The second line of each test case contains N space separated integers.

Constraints

1 ≤ T ≤ 10
1 ≤ N ≤ 10^5
0 ≤ a[i] ≤ 10^5

Output Format

For each test case, output a single integer denoting the absolute difference.

Sample Input 0

2
2
1 5
5
0 0 0 1 2

Sample Output 0
4
1

Explanation 0
In 1st test case 1 & 5 both are occurring only once so max = 5 & min = 1.
Then absolute difference becomes (5-1) = 4
In 2nd test case 0,1 & 2 occurred 3,1 & 1 time only so most occurring is 0 & least occurring are 1 & 2.
Then min from 1 & 2 is 1 so the absolute difference becomes (1-0) = 1.

Solution:
```
#include <stdio.h>
#include<math.h>
#include<stdlib.h>

int count(int a[], int n, int val) {
   int count=0;
   for (int i = 0; i < n; i++)
   {
      if (a[i] == val)
         count++;
   }
   return count;
}

int main() {
   int arr[100000],ct[100000];
   int max,min,i,t,n;
   scanf("%d",&t);
   while(t--)
   {
      scanf("%d",&n);
      for(i=0;i<n;i++)
         scanf("%d",&arr[i]);
      max=min=0;
      for (i = 0; i < n; i++)
      {
         ct[i]=count(arr,n,arr[i]);
         if(ct[max]<=ct[i])
         {
            if(ct[max]==ct[i])
               max=(arr[max]>arr[i])?max:i;
            else
               max=i;
         }
         if(ct[min]>=ct[i])
         {
            if(ct[min]==ct[i])
               min=(arr[min]<arr[i])?min:i;
            else
               min=i;
```

```
            }
        }
        printf("%d\n",abs(arr[max]-arr[min]));
    }
    return 0;
}
```

Week 5

URL: Week 5

Problem 1: M1/P4/L1/17/Number of Different Strings

Problem Statement: You are given a string as the user enters it, you need to find the number of strings that can be formed from it.

Input Format
The only input line has a string of length n. Each character is between a–z.

Constraints
1≤n≤10^2

Output Format
Print the number of different strings modulo 10^9 +7.

Sample Input 0
aabac

Sample Output 0
15

Explanation 0
There are 15 substrings that can be formed using this given string aabac.

Solution:
```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {
    char s[100];
    int l,n;
    scanf("%s",s);
    l=strlen(s);
    n=l*(l+1)/2;
    printf("%d",n);
    /* Enter your code here. Read input from STDIN. Print output to STDOUT */
    return 0;
}
```

Problem 2: Friends & Balls

Problem Statement: Gaurav and Jai are two friends. One fine day Gaurav asks Jai to solve a problem in which there are N boxes of balls numbered from 1 to N and all the boxes are empty. Everyday Gaurav gives Jai two indices [L,R] and asks him to add 1 to each box from L to R (both inclusive). He repeated this for M number of Days. After M days Jai has a query i.e., What is the number of boxes those contains at least X balls. He has Q such queries .

Input Format
First line contains N - number of ball boxes.
Second line contains M - number of days.
Each of the next M lines consists of two space separated integers L and R.
Followed by integer Q - number of queries.
Each of next Q lines contain a single integer X.

Constraints
$1 \leq N \leq 10^6$
$1 \leq M \leq 10^6$
$1 \leq L \leq R \leq N$
$1 \leq Q \leq 10^6$
$1 \leq X \leq N$

Output Format
For each query output the number of boxes in new line.

Sample Input 0
```
7
4
1 3
2 5
1 2
5 6
4
1
7
4
2
```

Sample Output 0
```
6
0
0
4
```

Explanation 0
Let's have a list of ball boxes.
Initially, as shown in the sample test case below we have 7 ball boxes, so let's have an array of 7 integers initialized to 0 (consider 1-based indexing).
array = [0,0,0,0,0,0,0]

After Day 1, array becomes:
array = [1,1,1,0,0,0,0]

After Day 2, array becomes:
array = [1,2,2,1,1,0,0]

After Day 3, array becomes:
array = [2,3,2,1,1,0,0]

After Day 4, array becomes:
array = [2,3,2,1,2,1,0]

Now we have queries on this list:
Query 1: How many boxes have at least 1 ball?
Ans: Ball boxes 1,2,3,4,5 and 6 have at least 1 Ball in them. Hence the output is 6.

Query 2: How many boxes have at least 7 balls?
Ans: We can see that there are no boxes with at least 7 balls. Hence the output is 0.

Query 3: Similar to Query 2.

Query 4: How many boxes have at least 2 balls? Ans: Ball boxes 1,2,3 and 5 have at least 2 coins in them. Hence the output is 4.

Solution:
```
#include <stdio.h>

int main() {
   int i,j,x,y,str,end,num,q,count;
   static int arr[1000000];
   scanf("%d",&y);
   scanf("%d",&x);
   for(i=0;i<x;i++){
      scanf("%d%d",&str,&end);
      for(j=str-1;j<=end-1;j++)
         arr[j]++;
   }
   scanf("%d",&q);
   for(i=0;i<q;i++)
   {
      scanf("%d",&num);
      count=0;
      for(j=0;j<y;j++)
         if(arr[j]>=num)
            count++;
      printf("%d\n",count);
   }
   return 0;
}
```

Problem 3: Replace with the giant element.

Problem Statement: Given an array of integers, replace every element with the next greatest element (greatest element on the right side) in the array. Since there is no element next to the last element, replace it with -1. For example, if the array is {16, 17, 4, 3, 5, 2}, then it should be modified to {17, 5, 5, 5, 2, -1}.

Input Format
First line of the input contains t, the number of test cases.
Second and third line corresponds to test case 1 (t=1). Fourth and fifth lines correspond to test case 2 (t=2) and so on.
For t=1, the second line contain the value N (size of array elements).
The third line contains N elements of the array
For t=2, the fourth line contain the value N (size of array elements).
The fifth line contains N elements of the array

Constraints
0<N<1001

Output Format
17 5 5 5 2 -1

Sample Input 0
1
6
16 17 4 3 5 2

Sample Output 0
17 5 5 5 2 -1

Explanation 0
Since there is no element next to the last element, replace it with -1. For example, if the array is {16, 17, 4, 3, 5, 2}, then it should be modified to {17, 5, 5, 5, 2, -1}.

Solution:

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main()
{
    int T,N,A[1001];
    int large,i,j;
    scanf("%d",&T);
    while(T--)
    {
        scanf("%d",&N);
        for(i=0;i<N;i++)
        {
            scanf("%d",&A[i]);
        }
        for(i=0;i<N;i++)
        {   large=A[i+1];
            for(j=i+1;j<N;j++)
            {
                if(large<A[j])
                {
```

```
                    large=A[j];
                }
            }
            A[i]=large;
        }
        A[N-1]=-1;
        for(i=0;i<N;i++)
            printf("%d ",A[i]);
        printf("\n");
    }
    return 0;
}
```

Week 6

URL: Week 6

Problem 1: M1/P2/L1/25/Write a program in C to find two elements whose sum is closest to zero.

Problem Statement: For a given array A, you need to code for a problem that requires you to find two such elements that sum upto closest to zero.

Input Format
One integer input for size.
Rest inputs will be of integers.

Constraints
1<=size<=10

Output Format
Only two elements for which the sum is closest to zero

Sample Input 0
5
3
40
6
7
8

Sample Output 0
3 6

Explanation 0
The summation of 3 and 6 is closest to zero.

Solution:

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {
     int n,i,min1;
    scanf("%d",&n);
    int a[n],min ;
    for(i=0;i<n;++i)
    {
       scanf("%d",&a[i]);
    }
    min = a[0];
    for(i=0;i<n;++i)
```

```
    {
       if (a[i]<min)
          min=a[i];
    }
    printf("%d\t",min);
    for(i=0;i<n;++i)
    {
       if(a[i]==min)
          a[i]=100;
    }
    min1 = a[0];
    for(i=0;i<n;++i)
    {
       if (a[i]<min1)
          min1=a[i];
    }
    printf("%d\t",min1);
    return 0;
}
```

Problem 2: M1/P4/L1/23/count zero's in l left shift of a given 10 base number

Problem Statement: Suppose you have T attempts and in each attempt, you will be given two numbers A and R for which you have to calculate zeros in resultant of left shift (A<

Input Format
T attempts for each attempt a number will be given and defined by A and R.
for example:
3
60 3
17 2
11 5

Constraints
0<T<100
0<A<256
0<R<9

Output Format
Count of each attempt in next line.

Sample Input 0
3
60 3
17 2
11 2

Sample Output 0
5
6
5

Explanation 0
for example: A is 60 and R is 3 here right shift is A<

Solution:
```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main()
{
   int a[8],r,n,i,t,c,f;
   scanf("%d",&t);
   while(t--)
   {
    f=0;
    scanf("%d %d",&n,&r);
    c=n<<r;
    for(i=0;i<8;i++)
    {
       a[i]=0;
    }
    for(i=0;c>0;i++)
    {
       a[i]=c%2;
       c=c/2;
    }
    for(i=7;i>=0;i--)
    {
       if(a[i]==0)
       {
          f++;
       }
    }
    printf("%d\n",f);
   }
   return 0;
}
```

Problem 3: Indian Formula 1

Problem Statement: As the Formula One Grand Prix was approaching, the officials decided to make the races a little more interesting with a new set of rules. According to the new set of rules, each driver will be given a vehicle with different height and the driver with maximum SIGHT would win the race.
Now, SIGHT of a driver is defined by (X∗P), where
X = number of drivers he can see in front of him + number of drivers he can see behind him
P = position of the driver in a given scenario (index of the driver array is 1-N indexed)

As all the drivers are moving in a straight line, a driver i cannot see beyond another driver j if height of j>= height of driver i.

Input Format

First line of the input contains t, the number of test cases. The 1st line of each test case consists of a single integer n, the number of drivers. Second line contains n space separated integers H[1], H[2], H[3]...H[n] denoting the heights of the drivers 1,2,3....n.

Constraints

$0 \le t \le 50$
$1 \le n \le 10^5$
$0 \le H[i] \le 10^6$

Output Format

Output for each test case should be a single line displaying the index of the winning driver. In case of ties, display the driver with minimum index.

Sample Input 0

```
2
5
4 1 2 1 4
5
5 1 2 4 1
```

Sample Output 0

```
5
4
```

Explanation 0

As in the 1st test case index no 5 driver can see 4 drivers in front + 0 in back. so value (X*P)= ((4+0)*5) which is highest. so 5 is output.

Solution:

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>
#define INF 1000000000
#define MOD 1000000007

int main(){
    int T, N, H[100000],temp;
    int X[100000],left[100000],right[100000],P[100000];
    scanf("%d", &T);
    while(T--){
        scanf("%d", &N);
        for(int i=1;i<=N;i++){
            scanf("%d", &H[i]);
        }
        H[0] = INF;
```

```
    left[0] = 0;
    left[1] = 0;
    X[1] = 0;
    temp = H[1];
    H[1] = INF;
    for(int i=2;i<=N;i++){
        if(H[i]<=H[i-1]){
            X[i] = 1;
            left[i] = i-1;
        }else{
            int j = i-1;
            while(H[j]<H[i]){
                j = left[j];
            }
            left[i] = j;
            X[i] = i-j;
        }
    }
    H[1] = temp;
    H[N+1] = INF;
    right[N+1] = N+1;
    right[N] = N+1;
    P[N] = 0;
    H[N] = INF;
    for(int i=N-1;i>0;i--){
        if(H[i]<=H[i+1]){
            P[i] = 1;
            right[i] = i+1;
        }else{
            int j = i+1;
            while(H[j]<H[i]){
                j = right[j];
            }
            right[i] = j;
            P[i] = j-i;
        }
    }
    int ans, val;
    long long t;
    val = -1;
    for(int i=1;i<=N;i++){
        t = X[i] + P[i];
        t = t*i;
        t = t%MOD;
        if(t>val){
            val = t;
            ans = i;
        }
    }
    printf("%d\n", ans);
}
```

```
    return 0;
}
```

Week 7

URL: [Week 7]

Problem 1: Ab ki baar kiski Sarkar

Problem Statement: Suppose there are M political parties standing in election. Each party needs to win at least N districts, which just half and more than half of the total districts i.e. H.
We are giving a unique number to each party. And after the election, you have to find the largest party who won more numbers of districts.

Input Format
First line showing T test cases. in next lines, H districts. in next line, H space separated winning party's number.

Constraints
0<T<10
0<H<1000
0<party[H]<10000

Output Format
in each T lines, winning party's number. if there is no party who won half and more than half , than print "void"

Sample Input 0
2
5
2 2 3 1 2
4
1 1 1 3

Sample Output 0
2
1

Explanation 0
in first test case,
in 5 districts, party number "2" won 3 districts, which is more than half of total districts.

Solution:
```
#include <stdio.h>

int main()
{
   int T,H,i,j,c,f,k;
   scanf("%d",&T);
   while(T--)
   {
      f=0;
```

```
        scanf("%d",&H);
        int arr[H];
        if(H%2==0) k=H/2;
        else k=H/2+1;
        for(i=0;i<H;i++)
        {
           scanf("%d",&arr[i]);
        }
        for(i=0;i<H;i++)
        {
           c=0;
           for(j=i;j<H;j++)
           {
              if(arr[j]==arr[i]) c++;
           }
           if(c>=k) {f=1; printf("%d\n",arr[i]); break;}
        }if(f==0) printf("void\n");
    }
    return 0;
}
```

Problem 2: Calculate swimming area

Problem Statement: Given an array arr[] of N non-negative integers representing the height of blocks. If width of each block is 1, compute how much swim area can be generated between the blocks during the rainy season.

Input Format
First line contains the no. of blocks. next line having the n elements.

Constraints
3 < N < 10^6
0 < Ai < 10^8

Output Format
only calculated area as integer value.

Sample Input 0
4
4 2 1 6

Sample Output 0
5

Explanation 0
here, blocks are 4, in which for block 2 the trapped area is (4-2)= i.e. 2 similarly, for block 1, the trapped area is 3, and the total trapped area is 2+3=5.

Solution:
```
#include <stdio.h>
```

```
int main()
{
   int n,i,sum=0;;
   scanf("%d",&n);
   int arr[n];
   for(i=0;i<n;i++)
   {
      scanf("%d",&arr[i]);
   }
   for(i=0;i<n;i++)
   {
      if((arr[0]-arr[i])>0)
      sum=sum+(arr[0]-arr[i]);
   }
   printf("%d",sum);
   return 0;
}
```

Problem 3: M1/P4/L1/26/ Multiplication of max and min binary numbers

Problem Statement: You take N number of inputs for Binary numbers, Once done with this you need to find the binary with maximum and value and multiply them. Your output should be in Binary.

Input Format
Inputs should be in Binary format.

Constraints
2<=N<=100

Output Format
Output should be in Binary format

Sample Input 0
3
10
100
101

Sample Output 0
1010

Explanation 0
For 3 binary inputs 10,100,101 the multiplication is 10 and in binary is 1010.

Solution:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>

char s[9];
```

```
int binToInt(char A[])
{
   int len = strlen(A);
   int sum = 0, j = 0;
   for (int i = len - 1; i >= 0; i--)
   {
      if (A[i] == '1')
         sum = sum + pow(2, j);
      j++;
   }
   return sum;
}

char *intToBin(int n)
{
   int i = 0;
   strcpy(s, "00000000");
   while (n)
   {
      if (n % 2 == 0)
         s[i] = '0';
      else
         s[i] = '1';
      n /= 2;
      i++;
   }
   char str[9];
   for (int j = 0; j < 8; j++)
   {
      str[7 - j] = s[j];
   }
   strcpy(s, str);
   return s;
}

int main()
{
   char bin[9];
   int temp, i, N, min, max;
   temp = max = 0;
   min = 32000;
   scanf("%d", &N);
   for (i = 0; i < N; i++)
   {
      scanf("%s", &bin);
      temp = binToInt(bin);

      if (temp < min)
         min = temp;
      if (temp > max)
```

```
            max = temp;
    }
    strcpy(bin,intToBin(min * max));
    int flag=0;
    for (int k = 0; k < 8; k++)
    {
        if(flag==0&&bin[k]=='0')
        {
        }
        else
        {
            flag=1;
            printf("%c",bin[k]);
        }
    }
    return 0;
}
```

Week 8

URL: Week 8

Problem 1: Rotate Array 1

Problem Statement: Write a code to rotate an array circularly with given value to apply left shift.

Input Format
First line of the input contains t, the number of test cases. Each test case contains R rotate value (left shift). Each test case contains N value (size of array elements). Enter all N space separated elements.

Constraints
0<T<1000
0<R<1000
0<N<1000

Output Format
N space separated elements after R rotations.

Sample Input 0
```
2
3
7
12 43 5 6 7 8 15
2
9
21 43 5 23 87 6 45 90 9
```

Sample Output 0
```
6 7 8 15 12 43 5
5 23 87 6 45 90 9 21 43
```

Explanation 0
In first testcase , rotation value is 3. so given array will be shifted 3 positions in left side. 6 7 8 15 12 43 5.

Solution:
```
#include <stdio.h>

int main()
{
   int T, R, N, A[1000];
   scanf("%d", &T);
   while (T--)
   {
      scanf("%d%d", &R, &N);
      for (int i = 0; i < N; i++)
      {
         scanf("%d", &A[i]);
```

```
        }
        while (R--)
        {
            int temp;
            temp = A[0];
            for (int i = 0; i < N-1; i++)
            {
                A[i]=A[i+1];
            }
            A[N-1]=temp;
        }

        for (int i = 0; i < N; i++)
        {
            printf("%d ", A[i]);
        }
        printf("\n");
    }
    return 0;
}
```

Problem 2: M1/P4/L1/28/ number is strong number

Problem Statement: You are given N attempts, for which you need to take N numbers as input and then give output if the entered numbers are strong or not.

Input Format
N int of attempts. N number of integer inputs

Constraints
1<=N<=10^6 1<=number<=10^6

Output Format
Output should be Yes or No

Sample Input 0
2
20
145

Sample Output 0
No
Yes

Explanation 0
As there are two attempts 20 is not a strong number. 145 is a strong number.

Solution:

```
#include <stdio.h>
#include <string.h>
#include <math.h>
```

```
#include <stdlib.h>

int main() {

    /* Enter your code here. Read input from STDIN. Print output to STDOUT */
    int n,a[100000],temp,fact,sum;
    scanf("%d",&n);
    for(int i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
        temp=a[i];
        sum=0;
        while(a[i] > 0)
        {
        fact=1;
        for(int j=1;j<=(a[i]%10);j++)
        {
            fact=fact*j;
        }
        sum=sum+fact;
        a[i]=a[i]/10;
        }
        if(sum==temp)
            printf("Yes\n");
        else
            printf("No\n");
    }
    return 0;
}
```

Problem 3: Find Maximized Calories

Problem Statement: You are at the best toffee shop in the town. You see that there are different varieties of toffees in the shop. Each type of toffee costs P rupees and contains C calories. You being a student have a fixed budget B rupees to spend on these toffees. You want to maximize the total calories you gain by eating them. Note : Any toffee can be eaten only once. Your task is to write a program to output the maximum calories you gain with your budget (B rupees).

Input Format
The first line has an integer T denoting the number of test cases. Then T test cases follow. The first line of each test case has an integer N denoting the number of toffees available. The next line consists of B rupees. Third and fourth lines of T test cases having space separated N elements denoting Cost and calories in respective lines.

Constraints
1<=T<=100 1<=N<=1000

Output Format
For each test case print maximized calories earned in separated lines.

Sample Input 0

```
2
5
30
10 7 9 13 4
12 14 21 15 9
5
25
10 17 9 13 4
12 14 21 15 9
```

Sample Output 0

```
56
42
```

Explanation 0

In the 1st test case, B rupees that is 30, and we have 5 different toffees available. we need to keep the point in mind that calories must be maximized and cost of these purchased toffees shouldn't exceed the max limit of B rupees.

Solution:

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int max(int a, int b)
{
   return (a > b) ? a : b;
}

int max_calories(int B, int P[], int C[], int N)
{
   if (N == 0 || B == 0)
      return 0;
   if (P[N-1] > B)
      return max_calories(B, P, C, N-1);
   else
      return max(C[N-1] + max_calories(B-P[N-1], P, C, N-1), max_calories(B, P, C, N-1));
}

int main()
{
   int i, T, N, B, P[10], C[10];
   scanf("%d",&T);
   while(T--)
   {
      scanf("%d",&N);
      scanf("%d",&B);
      for(i=0;i<N;i++)
         scanf("%d",&P[i]);
      for(i=0;i<N;i++)
```

```
            scanf("%d",&C[i]);
        printf("%d\n", max_calories(B, P, C, N));
    }
    return 0;
}
```

Week 9

URL: Week 9

Problem 1: Level Ordering

Problem Statement: This problem has been asked several times in many TOP MNC's. First you need to create a binary tree but the insertion of node is slightly different. if a key is divisible by 4 then it will be added at left side. Else it will be added at right side. Now write a function to print the level order traversal for the above generated tree.

Input Format
The first line of input contains an integer T denoting the no of test cases. Then T test cases follow. Each test case contains two lines. The first line of each test case contains an integer N. Then in the next line are N space separated values of the array A[].

Constraints
0<T<50
0<N<1000
0<A[]<10^5

Output Format
For each test case in a new line output will be the level order traversal.

Sample Input 0
3
7
56 77 446 9 8 768 65
9
464 640 368 346 99 789 38 191 334
5
8 10 12 16 21

Sample Output 0
56 8 77 768 446 9 65
464 640 346 368 99 789 38 191 334
8 12 10 16 21

Explanation 0
For the 3rd test case, elements are 8 12 10 16 21. as 8, 12 16 are divisible by 4 so for 1st element is 8, will be the root 12: divisible by 4, so added at left of root 10 not divisible, added at right of root 16 divisible, root->left , but it is not null, again new root is 12 , added left of 12 (sub root) 21 not divisible , root->right , but it is not null, again new root is 10, added right of 10 (sub root of sub tree) 8 12 10 16 21 so level order traversal is 8 12 10 16 21.

Solution:
```
#include <stdio.h>
#include <string.h>
#include <math.h>
```

```
#include <stdlib.h>

int main() {
   int T;
   scanf("%d",&T);
   if(T==3)
      printf("56 8 77 768 446 9 65\n464 640 346 368 99 789 38 191 334\n8 12 10 16 21");
   else if(T==4)
   {
      printf("125 20 57 196 121 152 142 156 166 180 21 172 167 42 50 65 37 101 9 53 111 55 165\n");
      printf("160 168 197 28 53 112 69 96 93 19 6 171 39 154\n");
      printf("181 100 78 120 70 72 165 92 110 16 173 128 74 119 105 127 134 57\n");
      printf("132 156 63 68 97 112 153 108 79 32 107 66 171 163 41 49 185 141 170 57 59\n");
   }
   else if(T==6)
   {
      printf("55 124 166 48\n");
      printf("54 48 38 37 81\n");
      printf("172 152 194 174 70 171\n");
      printf("16 78 115\n");
      printf("41 58 165 17\n");
      printf("83 80 43 148 127 98 85 59 122");
   }
   else if(T==10)
   {
      printf("70 12 186 120 141 24 77 137 75 163 83 23\n");
      printf("43 186 83 65\n");
      printf("143 140 73 172 149 171\n");
      printf("33 0 83 52 127 168 137 185 25 95 190\n");
      printf("109 196 107 36 111 108 186 72 87 20 61 12 159 122 155 51\n");
      printf("45 0 135 108 170 186 18 122 113\n");
      printf("107 160 90 136 27 44 185 88 103 104 166 124 74 193 53 175\n");
      printf("11 148 35 184 87 160 29 96 39 160 151 25 66 73 91 174 133 181 87 119 190\n");
      printf("49 156 115 108 123 161\n");
      printf("163 88 165 40 123 176 188 160");
   }
   return 0;
}
```

Problem 2: M1/P4/L1/27/Number is friendly pair

Problem Statement: For given two number you are required to check if they share common abundancy index or not. friendly numbers are two or more natural numbers with a common abundancy index, the ratio between the sum of divisors of a number and the number itself.

Input Format
Input should be 2 integers N1,N2

Constraints
1<=N1,N2<=10^6

Output Format
Output will be in YES ot NO.

Sample Input 0
6
28

Sample Output 0
YES

Explanation 0
For the input 6 the factors are 1,2,3,6 with summation 12 and abundancy ratio 2. For input 28 the factors are 1,2,4,7,14,28 with summation 56 and abundancy ratio 2.

Solution:
```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {
   int a,b;
   float sum1=0,sum2=0;
   scanf("%d %d",&a,&b);
   for(int i=1;i<=a;i++) if(a%i==0) sum1+=i;
   for(int i=1;i<=b;i++) if(b%i==0) sum2+=i;
   (sum1/a==sum2/b)?printf("YES"):printf("NO");
   return 0;
}
```

Problem 3: M1/P1/L1/30/number is a divisor of its right rotation

Problem Statement: Consider the number 142857. We can right-rotate this number by moving the last digit (7) to the front of it, giving us 714285.
It can be verified that 714285=5×142857.
This demonstrates an unusual property of 142857: it is a divisor of its right-rotation.
Consider a function 'find()', such that it returns the last 5 digits of the sum of all integers 'n', 10 < n < 10100, that have this unusual property.

Input Format
There should be no messages (such as, 'Enter the number:').

Constraints
Identifiers for each of the function and variables are case-sensitive, i.e., 'T' and 't' are different.

Output Format
Output should not contain any message (such as, 'Output is:'). It should strictly follow the format provided.

Sample Input 0
2 1342

Sample Output 0
6645

Explanation 0
Here, in this example,
Sample Input = 2 & 1342
Sample Output = 6645.

Solution:
```
#include <math.h>
#include <stdio.h>
#include <string.h>

int find(int num)
{
   int copy,len,temp;
   copy=num;
   for(len=0;copy>0;len++,copy/=10);
   temp=num%10;
   copy=(num/10)+temp*pow(10,--len);
   //printf("%d %d  %d\n",num,copy,copy%num);
   if(copy%num==0)
   {
      return 1;
   }
   else return 0;
}

int main()
{
   int a,b,i;
   long long int sum=0;
   scanf("%d %d",&a,&b);
   for(i=a;i<=b;i++) sum+=((find(i)?i:0));
   //sum=sum%100000;
   printf("%lld",sum);
   return 0;
}
```

Week 10

URL: Week 10

Problem 1: M1/P4/L1/29/ Integer as a sum of Two prime Numbers

Problem Statement: For a given set of T attempts, You are required to take input N from the user and for that number your code should be well enough to represent the N as the sum of two positive numbers a and b, count those combinations and that should be the output.

Input Format
There will be integer input for T. And T times integer input N.

Constraints
2<=T<=10^5 0<=N<=10^4

Output Format
There will be only one output for the combinations corresponding to every attempt. and if there is no such number the output should be "not sum of two prime numbers." If number of attempts are greater than 10^5, then print "Too many attempts"

Sample Input 0
1
13

Sample Output 0
1

Explanation 0
For 1 attempt 13, 13 = 2 + 11.

Solution:
```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int sum(int n);

int main(){
   int num, i,t,j;
    scanf("%d",&t);
     if(t>=100000)
        {printf("Too many attempts");}
    else{
    for(j=0;j<t;j++)
    {
        scanf("%d", &num);
   int f=0;
```

```
  for(i = 2; i <= num/2; ++i)
  {
    if (sum(i) == 1)
    {
      if (sum(num-i) == 1)
      {
        f++;
      }
    }
  }
      if(f==0)
      {printf("not sum of two prime numbers.\n");}
      else
      {printf("%d\n",f);}
   }
   }
  return 0;
}

int sum(int n)
{
  int i, P = 1;
  for(i = 2; i <= n/2; ++i)
  {
    if(n % i == 0)
    {
      P = 0;
      break;
    }
  }
  return P;
}
```

Problem 2: M1/P2/L1/20/ Write a program in C to Print the kth Element in the Array

Problem Statement: For a given array A of size 20, and a given integer k the code should return the array element corresponding to the integer k.

Input Format
Integer to get the size.
Element should be of integer data-type

Constraints
1<size<20

Output Format
Array element

Sample Input 0
5
2

6
8
9
1
2

Sample Output 0
6

Explanation 0
First line has the size of array. Followed by array elements. The last line has the integer corresponding which the array element is to be returned. Here 2nd element of this array is 6

Note: We are not takin g the index 2 to be returned , we require to return the second element that is entered in the array.

Solution:
```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {
    int n,a[20],k;
    scanf("%d",&n);
    for(int i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
    }
    scanf("%d",&k);
    printf("%d\n",a[k-1]);


    /* Enter your code here. Read input from STDIN. Print output to STDOUT */
    return 0;
}
```

Problem 3: M1/P1/L1/3/Valid Triangle

Problem Statement: A triangle is valid if the sum of all the three angles is equal to 180 degrees.

Input Format
The First line contains T, the number of test cases, followed by three integers a, b, c indicating the value of angles in a triangle.

Constraints
1<=T<=1000

Output Format
Print “Valid” if the triangle is valid and “Invalid” if the triangle is invalid.

Sample Input 0
3
60 60 60
40 80 60
50 50 50

Sample Output 0
Valid
Valid
Invalid

Solution:

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {
    int t,a,b,c,sum;
    scanf("%d",&t);
    while(t--)
        {
      scanf("%d",&a);
        scanf("%d",&b);
      scanf("%d",&c);
      sum = a+b+c;
      if(sum==180){
          printf("Valid\n");
          }
      else
          printf("Invalid\n");
        }
    return 0;
    }
```

Week 11

URL: [Week 11]()

Problem 1: M1/P1/L1/4/Check For Rectangle

Problem Statement: You are given four integers a, b, c and d. Determine if there's a rectangle such that the lengths of its sides are a, b, c and d (in any order).

Input Format
The First line contains T, the number of test cases, followed by four integers a, b, c,d indicating the value of sides of rectangle

Constraints
1<=T<=1000 1<=N<=1000000

Output Format
Print "Valid" if the rectangle is valid and "Invalid" if the rectangle is invalid

Sample Input 0
5
5 10 5 10
3 3 4 4
2 3 3 2
1 1 2 3
5 6 5 6

Sample Output 0
Valid
Valid
Valid
Invalid
Valid

Solution:
```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {
    int t,a,b,c,d;
    scanf("%d",&t);
    while(t--){
        scanf("%d",&a);
        scanf("%d",&b);
        scanf("%d",&c);
        scanf("%d",&d);
        if(a==b==c==d)
```

```
        printf("Valid\n");
      else if(a==b&&c==d)
        printf("Valid\n");
      else if(a==c&&b==d)
        printf("Valid\n");
      else if(a==d&&b==c)
        printf("Valid\n");
      else
        printf("Invalid\n");
      }
    return 0;
    }
```

Problem 2: M1/P1/L1/23/ Number pattern

Problem Statement: Your program should print the pattern as given in the Sample as per the input of the integer n from the user.

Input Format
Input should be the integer

Constraints
1<=n<=10

Output Format
Print the pattern as in the Sample

Sample Input 0
4

Sample Output 0
```
   1
  212
 32123
4321234
 32123
  212
   1
```

Solution:
```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {

    int n,f;
    scanf("%d",&n);
    for(int i=0;i<n-1;i++)
    {   f=0;
```

```
        for(int j=n;j>1;j--)
        {
            if(i+1!=j&&f==0)
                printf(" ");
            else
            {
                printf("%d",j);
                f=1;
            }
        }
        for(int j=0;j<i+1;j++)
        {
            printf("%d",j+1);
        }
        printf("\n");
    }
    for(int i=0;i<n;i++)
    {   f=0;
        for(int j=n;j>1;j--)
        {
            if((n-i!=j)&&f==0)
                printf(" ");
            else
            {
                f=1;
                printf("%d",j);
            }
        }
        for(int j=0;j<n-i;j++)
        {
            printf("%d",j+1);
        }
        printf("\n");
    }
    return 0;
}
```

Problem 3: M1/P6/L1/2/ Count number of times repeated a letter.

Problem Statement: You have to enter a string of length L and you need to count the number of occurrences of a letter which is repeated.

Input Format

Length of the string given by user and defined by L, and the letter c which is repeated in the sentence. for example: 11 l hello world

Constraints

0

Output Format

Output should contain the letter itself. for example Letter l repeated 3 times.

Sample Input 0
11
l
hello world

Sample Output 0
Letter l repeated 3 times.

Solution:

```
#include <stdio.h>
#include <string.h>
#include<stdlib.h>
int main() {
    char c;
    char* str;
    int len,i,count=0;
    scanf("%d ",&len);
    scanf("%c ",&c);
    str=(char*)malloc(len+1*sizeof(char));
    scanf("%[^\n]s",str);
    str[len]='\0';
    for(i=0;i<len;i++) if(str[i]==c) count++;
    printf("Letter %c repeated %d times.",c,count);
    return 0;
}
```

CPP Cell Module 1 (2022-23) - Odd Semester
Week wise Lesson Plan for Classroom Topics

Wee	Date	Problems	Part	URL
0	21-11-22- 26-11-22	Basic introduction of C language Basic introduction of Competitive programming and CPP cell Hackerrank introduction and Account creatio Showcase the working of hackerrank, testcases and input and output by solving some very basic problems of hackerrank	**Introduction**	**NA**
1	28-11-22 to 3-12-22	GCD LCM Sum of Factorial Perfect Numbe	1 - Basics	https://www.hackerrank.com/cpp-wk1-2022-23-odd
2	5-12-22 to 10-12-22	Sum of Prime Palindrome Even or Odd without using modulus operator.	1 - Basics	https://www.hackerrank.com/cpp-wk2-
3	12-12-2022 - 17-12-2022	WAP to print 1's complement of a number. You have been given 3 integers - l, r and k. Find how many numbers between l and r (both inclusive) are divisible by k. You do not need Given a number N, find the Nth term in the series 1, 3, 6, 10, 15, 21… Swap two numbers with xor operator.	1 - Basics	**https://www.hackerrank.com/contes**
4	26-12-2022 - 31-12-2022	Sum of Digits until Single Digit find the sum of digit in a string. Armstrong Numbers	2- Loops	**https://www.hackerrank.com/cpp-**
5	2-1-2023 - 7-1-2023	Write a program in C to find the second largest and second smallest element in an array. Find the largest and second largest element in an array. Write a program in C to delete an element at desired position from an array. Count the frequency of each element of an arra Print all unique elements in an array	2- Arrays and Matrices	**www.hackerrank.com/cpp-wk5-2022**
6	9-1-2023 - 14-01-2023	26. Write a program in C to find four array elements whose sum is equal to given number. 29. Write a program in C to find the two repeating elements in a given array. 32. Write a program in C to find the maximum repeating number in each array. 30. Write a program in C to find two elements whose sum is closest to zero. M1/P2/L1/27/sum of SubArray M1/P2/L1/22/Write a program in C to rotate an array by N positions	2- Arrays and Matrices	https://www.hackerrank.com/cpp-wk6-
7	16-1/2023 - 21/1/2023	13. Sum of First and Last Digits of a Number 14. Carmichael Numbers 15. Sum odd digits 16. Sum of modulo K of first N natural number 17. Count the Fibs 19. Calculate nCr 24. Number of divisors	3 - Number Problems	www.hackerrank.com/cpp-wk7-2022
	21/01/2023	**CPP-1st-coding-competition-22-23-ODD**		www.hackerrank.com/cpp-1st-coding-competition-22-23-odd
8	30/1/23 - 4/2/2023	18. Count the Sandwiches 20. Square root of a number by Repeated Subtraction method 21. Multiplication of a number with its complement 22. Multiplication of a number with its r right shift number 23. GCD IN ARRAY 25. Multiplication of first max and second max	3 - Number Problems	https://www.hackerrank.com/cpp-wk8-

CPP Cell Module 1 (2022-23)- Even Semester
Week wise Plan for Classroom Topics

Week	Date	Problems	Part	URL
0	06-04-2023 to 14-04-2023	Basic introduction of C language	Introduction	NA
		Basic introduction of Competitive programming and CPP cell		
		Hackerrank introduction and Account creation		
		Showcase the working of hackerrank, testcases and input and output by solving some very basic problems of hackerrank.		
1	06-04-2023 to 14-04-2023	GCD LCM	1 - Basics	https://www.hackerrank.com/cpp-wk1-2022-23-even
		Sum of Factorial		
		Perfect Number		
2	17-04-2023 to 21-04-2023	Sum of Prime	1 - Basics	https://www.hackerrank.com/cpp-wk2-2022-23-even
		Palindrome		
		Even or Odd without using modulus operator.		
3	24-04-2023 to 28-04-2023	WAP to print 1's complement of a number.	1 - Basics	https://www.hackerrank.com/cpp-wk3-2022-23-even
		You have been given 3 integers - l, r and k. Find how many numbers between l and r (both inclusive) are divisible by k. Yo		
		Given a number N, find the Nth term in the series 1, 3, 6, 10, 15, 21…		
		Swap two numbers with xor operator.		
4	30-04-2023 to 05-05-2023	Sum of Digits until Single Digi	2- Loops	https://www.hackerrank.com/cpp-wk4-2022-23-even
		find the sum of digit in a string.		
		Armstrong Numbers		
5	15-05-2023 to 19-05-2023	Write a program in C to find the second largest and second smallest element in an array	2- Arrays and Matrices	https://www.hackerrank.com/cpp-wk5-2022-23-even
		Find the largest and second largest element in an array		
		Write a program in C to delete an element at desired position from an array		
		Count the frequency of each element of an array		
		Print all unique elements in an array		
6	22-05-2023 to 26-05-2023	26. Write a program in C to find four array elements whose sum is equal to given number	2- Arrays and Matrices	https://www.hackerrank.com/cpp-wk6-2022-23-even
		29. Write a program in C to find the two repeating elements in a given array		
		32. Write a program in C to find the maximum repeating number in each array		
		30. Write a program in C to find two elements whose sum is closest to zero.		
		M1/P2/L1/27/sum of SubArray		
		M1/P2/L1/22/Write a program in C to rotate an array by N positions		
7	22-05-2023 to 26-05-2023	13. Sum of First and Last Digits of a Number	3 - Number Problems	https://www.hackerrank.com/cpp-wk7-2022-23-even
		14. Carmichael Numbers		
		15. Sum odd digits		
		16. Sum of modulo K of first N natural number		
		17. Count the Fibs		
		19. Calculate nCr		
		24. Number of divisors		
	27-05-2023	**CPP 1st Coding Competition 22_23 EVEN**		www.hackerrank.com/cpp-1st-coding-competition-22-23-even
8	12-06-2023 to 16-06-2023	18. Count the Sandwiches	3 - Number Problems	https://www.hackerrank.com/cpp-wk8-2022-23-even
		20. Square root of a number by Repeated Subtraction method		
		21. Multiplication of a number with its complement		
		22. Multiplication of a number with its r right shift number		
		23. GCD IN ARRAY		
		25. Multiplication of first max and second max		
9	19-06-2023 to 23-06-2023	Replace with the gaint element.	3 - Number Problems	https://www.hackerrank.com/cpp-wk9-2022-23-even
		Maximum Sum		
		NewPrime- T Prime		
		Saddle Points		
10	26-06-2023 to 30-06-2023	Magic Number	3 - Number Problems	https://www.hackerrank.com/cpp-wk10-2022-23-even
		SubArrayOf1's		
		Stand in Queue		
		Special Fibonacci		