LAB WORKBOOK

21SC1202- DATA STRUCTURES

FED : BES-I

Team of FED : BES-I
K L E F | DATA STRUCTURES 21SC1202

Koneru Lakshmaiah Education Foundation
(Category -1, Deemed to be University estd. u/s. 3 of the UGC Act, 1956)
Accredited by **NAAC** as '**A++**' ❖Approved by AICTE ❖ ISO 9001-2015 Certified
Campus: Green Fields, Vaddeswaram - 522 302, Guntur District, Andhra Pradesh, INDIA.
Phone No. 08645 - 350200; www.klef.ac.in; www.klef.edu.in; www.kluniversity.in
Admin Off: 29-36-38, Museum Road, Governorpet, Vijayawada - 520 002. Ph: +91 - 866 - 3500122, 2577715, 2576129.

LABORATORY WORKBOOK

Name of the Student :	
Register Number :	
Year :	**I**
Semester :	**Second**
Section :	
Name of the Faculty :	

INDEX

Exp.No	Date	Title of the Programs	Remarks & Signature
1		Develop a program to implement below sorting techniques a. Insertion Sort b. Shell Sort	
2		Develop a program to implement below sorting technique a. Quick Sort	
3		Develop a program to implement below sorting technique a. Merge Sort	
4		List Implementation - a. Singly Linked List	
5		Circular Linked List a. Circular Linked List b. Doubly Linked List	
6		Linked Implementation of – a. Stack	
7		Linked Implementation of – a. Queue	
8		Stack Applications – a. Infix to Postfix Expression Conversion b. Postfix Expression Evaluation	
9		Hashing – a. Separate chaining b. Open Addressing	
10		Binary search Tree a. Creation of BST b. Traversal – In-order, Pre-order, Post-order	
11		AVL Tree – Self Balancing Tree	
12		Graphs – a. BFS b. DFS	

Faculty In-charge **Course Coordinator** **HOD/BES-1**

Table of contents

Lab -7	Pre requisite for Queue	
	Pre-Lab Task	
	In- Lab Task	
	Post- Lab Task	
	Skill Session	
Lab -8	Pre requisite for Stack Applications	
	Pre-Lab Task	
	In- Lab Task	
	Post- Lab Task	
	Skill Session	
Lab -9	Pre requisite for Hashing	
	Pre-Lab Task	
	In- Lab Task	
	Post- Lab Task	
	Skill Session	
Lab -10	Pre requisite for BST	
	Pre-Lab Task	
	In- Lab Task	
	Post- Lab Task	
	Skill Session	
Lab -11	Pre requisite for AVL Tree	
	Pre-Lab Task	
	In- Lab Task	
	Post- Lab Task	
	Skill Session	
Lab -12	Pre requisite for Graphs	
	Pre-Lab Task	
	In- Lab Task	
	Post- Lab Task	
	Skill Session	

Organization of the STUDENTS LAB WORKBOOK

The laboratory framework includes a creative element but shifts the time-intensive aspects outside of the Two-Hour closed laboratory period. Within this structure, each laboratory includes three parts: Pre-lab, In-lab, and Post-lab.

a. Pre-Lab

The Pre-lab exercise is a homework assignment that links the lecture with the laboratory period - typically takes 2 hours to complete. The goal is to synthesize the information they learn in lecture with material from their textbook to produce a working piece of software. Pre-lab Students attending a two-hour closed laboratory are expected to make a good-faith effort to complete the Pre-lab exercise before coming to the lab. Their work need not be perfect, but their effort must be real (roughly 80 percent correct).

b. In-Lab

The In-lab section takes place during the actual laboratory period. The First hour of the laboratory period can be used to resolve any problems the students might have experienced in completing the Pre-lab exercises. The intent is to give constructive feedback so that students leave the lab with working Pre-lab software - a significant accomplishment on their part. During the second hour, students complete the In-lab exercise to reinforce the concepts learned in the Pre-lab. Students leave the lab having received feedback on their Pre-lab and In-lab work.

c. Post-Lab

The last phase of each laboratory is a homework assignment that is done following the laboratory period. In the Post-lab, students analyze the efficiency or utility of a given system call. Each Post-lab exercise should take roughly 120 minutes to complete.

2021-22 EVEN SEMESTER LAB CONTINUOUS EVALUATION

S. No	Date	Experiment Name	Pre-Lab (15M)			In-Lab(30M)				Viva Voce (5M)	Total (50M)	Faculty Signature
			Logic (5M)	Implementation (5M)	Execution(5M)	Logic (10M)	Execution (10M)	Result (5M)	Analysis (5M)			
1												
2												
3												
4												
5												
6												
7												
8												

2021-22 EVEN SEMESTER LAB CONTINUOUS EVALUATION

S. No	Date	Experiment Name	Pre-Lab (15M)			In-Lab(30M)				Viva Voce (5M)	Total (50M)	Faculty Signature
			Logic (5M)	Implementation (5M)	Execution(5M)	Logic (10M)	Execution (10M)	Result (5M)	Analysis (5M)			
9												
10												
11												
12												

Faculty In-Charge Course Coordinator HOD/BES-I

2021-22 EVEN SEMESTER SKILL CONTINUOUS EVALUATION

Sl No	Date	Experiment Name	Skill Session(50M)				Total (50M)	Faculty Signature
			LOGIC (20M)	EXECUTION (10M)	RESULT (10M)	ANALYSIS (10M)		
1								
2								
3								
4								
5								
6								
7								
8								

2021-22 EVEN SEMESTER SKILL CONTINUOUS EVALUATION

Sl No	Date	Experiment Name	Skill Session(50M)				Total (50M)	Faculty Signature
			LOGIC (20M)	EXECUTION (10M)	RESULT (10M)	ANALYSIS (10M)		
9								
10								
11								
12								

Faculty In-Charge Course Coordinator HOD/BES-I

FED : DEPARTMENT OF BASIC ENGINEERING SCIENCE – I SUBJECT CODE: 21SC1202
DATA STRUCTURES WORKBOOK

WEEK - 1

Lab Session:

Date of the Session: ___/___/___ **Time of the Session: ____to____**

Prerequisite:

- Divide and Conquer Algorithm
- Sorting
- swapping items in place
- Partitioning of the array.

Pre-lab Task:

1. In Indian battalion 35 – NCC Brigadier Vishal wants to sort his troop's corps in ascending order according to the corps height. So, help Brg.Vishal to arrange corps accordingly. (Imagine corps height as elements in array). Analyze the time complexity**(O(n log n))**

Hint: Insertion sort helps to sort array in best time complexity.

Program :-

2. Given a list of N array element apply shell sort.. The first line contains an integer, N, the number of elements in Array. The second line contains N space-separated integers. Print the array as a row of space-separated integers in each iteration.

Program :-

In-lab Task:

1) Implement Insertion Sort

Given a list of N array elements apply insertion sort on array.

Input Format

The first line contains the integer N, the size of the array .

The next line contains N space-separated integers .

Constraints

- 1<= N <= 1000
- -1000<= a[i]<= 1000

Output Format: Print the array as a row of space-separated integers each iteration .

https://www.hackerrank.com/contests/17cs1102/challenges/3-a-implement-insertion-sort

Program :-

2. Quick Sort

Given a list of N array elements apply Quick Sort.

Input Format

The first line contains an integer, N, the number of elements in Array.

The second line contains N space-separated integers.

Constraints

1<= N <= 1000

-1000 <= array elements <= 1000

Output Format

Print the array as a row of space-separated integers each iteration
https://www.hackerrank.com/contests/17cs1102/challenges/4a-quick-sort

Program :-

3. Merge Sort

Given a list of N array element apply Merge sort. Merge Sort is a Divide and Conquer algorithm. It divides input array in two halves, calls itself for the two halves and then merges the two sorted halves. The first line contains an integer, N, the number of elements in Array. The second line contains N space-separated integers. Print the array as a row of space-separated integers in each iteration.

https://www.hackerrank.com/contests/17cs1102/challenges/merge-sort-6

Program :-

Post-lab Task:

1. Noor and his pond

Noor is doing fish farming. There are N types of fish. Each type of fish has size(S) and eating factor(E). A fish with eating factor of E, will eat all the fish of size ≤E. Help Noor to select a set of fish such that the size of the set is maximized as well as they do not eat each other.

https://www.hackerearth.com/practice/algorithms/sorting/quick-sort/practice-problems/algorithm/noor-and-his-pond-760eabe0/

Program :-

2. Card game:

Two friends decided to play a very exciting online card game. At the beginning of this game, each player gets a deck of cards, in which each card has some strength assigned. After that, each player picks random card from his deck and they compare strengths of picked cards. The player who picked card with larger strength wins. There is no winner in case both players picked cards with equal strength.

First friend got a deck with *n* cards. The *i*-th his card has strength ai. Second friend got a deck with *m* cards. The *i*-th his card has strength bi.

First friend wants to win very much. So he decided to improve his cards. He can increase by *1* the strength of any card for *1* dollar. Any card can be improved as many times as he wants. The second friend can't improve his cards because he doesn't know about this possibility.

What is the minimum amount of money which the first player needs to guarantee a victory for himself?

https://www.hackerearth.com/practice/algorithms/sorting/quick-sort/practice-problems/algorithm/card-game-1-44e9f4e7/

Program :-

Skill Session:

1) Merge Sort: Counting Inversions

In an array, arr , the elements at indices i and j (where i<j) form an inversion if arr[i]>arr[j]. In other words, inverted elements arr[i] and arr[j] are considered to be "out of order". To correct an inversion, we can swap adjacent elements.

Explanation:

We sort the following d=2 datasets:

1.arr = [1,1,1,2,2] is already sorted, so there are no inversions for us to correct.

2.arr = [2,1,3,1,2]

[2,1,3,1,2]—1st swap-----→[1,2,3,1,2]—2nd swap---→[1,1,2,3,2] ---3rd swap--→[1,1,2,2,3]

https://www.hackerrank.com/challenges/ctci-merge-sort/problem?h_r=internal-search

Program :-

2. Insertion Sort - Part 1

One common task for computers is to sort data. For example, people might want to see all their files on a computer sorted by size. Since sorting is a simple problem with many different possible solutions, it is often used to introduce the study of algorithms.

https://www.hackerrank.com/challenges/insertionsort1/problem

Program :-

3. DESORT

Cheetah is too lazy to write the problem statement, so here's the question- Given n positive numbers you have to sort those numbers in descending order. Each element A_i in the list consists of at most 10^5 numbers of digits.
Note: There are no leading zeros in input.

https://www.codechef.com/problems/DSORT

Program :-

4. Frog Sort

There are NN frogs (numbered 11 through NN) in a line. For each valid ii, the ii-th frog is initially at the position ii, it has weight WiWi, and whenever you hit its back, it jumps a distance LiLi to the right, i.e. its position increases by LiLi. The weights of the frogs are pair wise distinct.
You can hit the back of each frog any number of times (possibly zero, not necessarily the same for all frogs) in any order. The frogs do not interfere with each other, so there can be any number of frogs at the same time at each position.

Your task is to sort the frogs in the increasing order of weight using the smallest possible number of hits. In other words, after all the hits are performed, then for each pair of frogs (i,j)(i,j) such that Wi<WjWi<Wj, the position of the ii-th frog should be strictly smaller than the position of the jj-th frog. Find the smallest number of hits needed to achieve such a state.

https://www.codechef.com/problems/FROGS

Program :-

5. Problem Sort:

Chef is organising a contest with PP problems (numbered 11 through PP). Each problem has SS subtasks (numbered 11 through SS).

The *difficulty* of a problem can be calculated as follows:

- Let's denote the score of the kk-th subtask of this problem by SCkSCk and the number of contestants who solved it by NSkNSk.
- Consider the subtasks sorted in the order of increasing score.
- Calculate the number nn of valid indices kk such that NSk>NSk+1NSk>NSk+1.
- For problem ii, the difficulty is a pair of integers (n,i)(n,i).

You should sort the problems in the increasing order of difficulty levels. Since difficulty level is a pair, problem aa is more difficult than problem bb if the number nn is greater for problem aa than for problem bb, or if a>ba>b and nn is the same for problems aa and bb.

https://www.codechef.com/problems/PROBLEMS

Program :-

6. Suffix Sort:

The match is set and your team is ready to win!
The team has NN players, each with the ability AiAi.
But you are not happy with the given playing order. You are wondering how to create a new order so that the player with lesser ability go earlier, i.e., in **non-decreasing order**.

In one move, you can break a suffix of **any** length of the array and attach it to the front. You can do the move as many times as you like.
For example, the Array A=[1,2,3,4,5,6]A=[1,2,3,4,5,6] can become A=[5,6,1,2,3,4]A=[5,6,1,2,3,4] in one move by breaking and reattaching the suffix of length 22.

For each test case, if the array can be sorted non-decreasingly, print YES and the **minimum** number of moves to do so. Otherwise, print NO.

https://www.codechef.com/problems/ICM0001

Program :-

(For Evaluator's use only)

Comment of the Evaluator (if Any)	Evaluator"s Observation Marks Secured:_____out of _____ Full Name of the Evaluator: Signature of the Evaluator Date of Evaluation:

Week-2

Lab Session:

Date of the Session: ___/___/___ **Time of the Session: ____to____**

Prerequisite:

- Divide and Conquer Algorithm
- Sorting
- swapping items in place
- Partitioning of the array.

Pre Lab Task:

1. Chris and Scarlett were playing a block sorting game where Scarlett challenged Chris that he has to sort the blocks which arranged in random order. And Scarlett puts a restriction that he should not use reference of first, median and last blocks to sort, and after sorting one block with reference to other block, for next iteration he must choose another block as the reference not the same block (random pivot). Now, Chris wants help from you to sort the blocks. He wanted to sort them in a least time. Help him with the least time complexity sorting algorithm.

Program :-

2. A group of 9 friends are playing a game, rules of the game are as follows: Each member will be assigned with a number and the sequence goes like e.g.: 7, 6, 10, 5, 9, 2, 1, 15, 7. Now they will be sorted in ascending order in such a way that tallest one will be sorted first. Now your task is to find the order of indices based on initial position of the given sequence and print the order of indices at the end of the iteration.

Program :-

In Lab Task

1. Implement Shell Sort:

Given a list of N array elements apply shell sort on array.

Input Format

The first line contains the integer N, the size of the array.
The next line contains N space-separated integers.

Constraints

• 1<= N <= 1000
• -1000<= a[i]<= 1000

Output Format

Print the array as a row of space-separated integers in each iteration.

https://www.hackerrank.com/contests/17cs1102/challenges/3b-implement-shell-sort

Program :-

3. Insertion Sort

You are given an array AA of size NN. Sort the array using Insertion Sort and print the sorted array.

https://www.codechef.com/DSCA2019/problems/NSECDS03

Program :-

4. Quick sort 1 – Partition

The previous challenges covered Insertion Sort, which is a simple and intuitive sorting algorithm with a running time. In these next few challenges, we're covering a *divide-and-conquer* algorithm called Quick sort (also known as *Partition Sort*). This challenge is a modified version of the algorithm that only addresses partitioning. It is implemented as follows:

https://www.hackerrank.com/challenges/quicksort1/problem

Program :-

Post Lab Task:

1. Missing Number

You are given an array *A*. You can decrement any element of the array by *1*. This operation can be repeated any number of times. A number is said to be missing if it is the smallest positive number which is a multiple of *2* that is not present in the array *A*. You have to find the maximum missing number after all possible decrements of the elements.

https://www.hackerearth.com/practice/algorithms/sorting/quick-sort/practice-problems/algorithm/biggest-cake-possible-6d5915e7/

Program :-

2. One-Sized Game:

Ladia and **Kushagra** are playing the **One-Sized Game**. The rules of this game are pretty simple. During the game's play, if at any point of time, the array is left with only one element, Ladia wins the game otherwise Kushagra wins.

Given an array consisting of N elements, print whether Ladia will win or Kushagra.

https://www.hackerearth.com/practice/algorithms/sorting/quick-sort/practice-problems/algorithm/one-sized-game/

Program :-

Skill Session

1. Sorting Tool

Chaitya is a great problem solver. He found that most of his problems could be solved by arranging them in a particular order. The problem input consists of **N**. numbers, smaller than or equal to **M**. Chaitya's analysis consists of sorting this sequence according to their occurrences.

The output must be sorted so that for two numbers **A** and **B**, **A** will appear before **B** if the number of times **A** appears in the original order is larger than the number of times **B** does. If the number of occurrences is equal, the number whose value appears sooner in the input should appear sooner in the sorted sequence.

Help Chaitya by creating a custom sorting tool.

https://www.codechef.com/problems/KJCP01

Program :-

2. Willows Sort

Willow, a 3 years old genius, thought that sorting is based on the last digit of a number. His comparison process is if the last digit of an integer AA is smaller than the last digit of another integer BB ,then AA<BB. And if the last digit of two integers is same then he checks the previous one and this way the comparison continues.

That means, if the digits of an integer AA is represented as a1a1,a2a2…anan and another integer BB as b1b1,b2b2…bmbm (n and m not necessary to be equal), AA will be smaller than BB if there is an ii such that AiAi < BiBi and for each valid jj > ii, AjAj = BjBj.

https://www.codechef.com/problems/WISORT

Program :-

3. Radix Sort

We have already studied about many sorting techniques such as Insertion sort, Bubble sort, Selection sort, Quick sort, Merge sort, Heap sort etc. Here I will talk about a different type of sorting technique which is called as "Radix Sort" and is probably the best sorting technique as far as time complexity is concerned.

Operations which are performed in Radix Sort is as follows:-

1) Do following for each digit **i** where **i** varies from least significant digit to the most significant digit.

2) Sort input array using counting sort (or any stable sort) according to the **i'th** digit.

https://www.codechef.com/problems/RDX

Program :-

5. Insertion Sort - Part 2:

In Insertion Sort Part 1, you inserted one element into an array at its correct sorted position. Using the same approach repeatedly, can you sort an entire array?

Guideline: You already can place an element into a sorted array. How can you use that code to build up a sorted array, one element at a time? Note that in the first step, when you consider an array with just the first element, it is already sorted since there's nothing to compare it to.

In this challenge, print the array after each iteration of the insertion sort, i.e., whenever the next element has been inserted at its correct position. Since the array composed of just the first element is already sorted, begin printing after placing the second element.

https://www.hackerrank.com/challenges/insertionsort2/problem

Program :-

5. Correctness and the Loop Invariant

In the previous challenge, you wrote code to perform an Insertion Sort on an unsorted array. But how would you prove that the code is correct? I.e. how do you show that for any input your code will provide the right output?

https://www.hackerrank.com/challenges/correctness-invariant/problem

Program :-

6. Merge Sorted Array

You are given two integer arrays nums1 and nums2, sorted in non-decreasing order, and two integers' m and n, representing the number of elements in nums1 and nums2 respectively.

https://leetcode.com/problems/merge-sorted-array/

Program :-

(For Evaluator's use only)

Comment of the Evaluator (if Any)	Evaluator"s Observation Marks Secured:_____out of _____ Full Name of the Evaluator: Signature of the Evaluator Date of Evaluation:

Week-3

Lab Session:

Date of the Session: ___/___/___ **Time of the Session: ____to____**

Prerequisite:

- Divide and Conquer Algorithm
- Sorting
- swapping items in place
- Partitioning of the array.

Pre Lab Task

1. Trace out the output of the following using Merge sort.10, 49, 32, 67, 45, 4, 7, 2, 1, 51, 78, 34, 89, 87, 36, 29, 3, 9, 11.

Solution:-

2. Trace the insertion sort algorithm to arrange the following data in ascending order 49, 32, 67, 45, 4, 2, 1, 78, 34, 89, 87.

Solution:-

In Lab Task

1. Quick Sort

You are given an array AA of size NN. Sort the array using Quick Sort and print the sorted array.

https://www.codechef.com/DSCA2019/problems/NSECDS06/

Program :-

2. Max Power

Given an array A having N distinct integers.

The power of the array is defined as:

- max(A[i]−A[j]) where 2≤i≤N
- for each *i*, *j* is the largest index less than *i* such that A[j]<A[i].

Let's say the array is {1,2,5}, then the power of the array is max((2−1),(5−2)) , which simplifies to max(1,3) which is equal to *3*.

Operation Allowed:

If you are allowed to choose any two indices *x* and *y* and swap A[x] and A[y], find out the maximum power that can be achieved.

https://www.hackerearth.com/practice/algorithms/sorting/quick-sort/practice-problems/algorithm/increasing-subsequence-fbb63e3c/

Program :-

3. Maximum Sum of Building Speed

You are the king of Pennsville where you have 2N workers. All workers will be grouped in association of size *2*,so a total of *N* associations have to be formed. The building speed of the ith worker is Ai. To make an association, you pick up *2* workers. Let the minimum building speed between both workers be *x*, then the association has the resultant building speed *x*.

You have to print the maximum value possible of the sum of building speeds of *N* associations if you make the associations optimally.

https://www.hackerearth.com/practice/algorithms/sorting/merge-sort/practice-problems/algorithm/maximum-sum-of-building-speed-00ab8996/

Program :-

Post Lab Task:

1. Benny and Gifts:

Little pig Benny has just taken a shower. Now she is going to buy some gifts for her relatives. But the problem is that Benny doesn't know how to reach to the gift shop. Her friend Mike has created a special set of instructions for her. A set of instructions is a string which consists of letters {'L', 'R', 'U', 'D'}.

For simplicity, let's assume that Benny is staying at point (0, 0) on the infinite plane. She consistently fulfills all the instructions written by Mike. Let's assume that now she is staying at point (X, Y). Then depending on what is the current instruction she moves in some direction:

- 'L' -- from (X, Y) moves to point (X, Y - 1)
- 'R' -- from (X, Y) moves to point (X, Y + 1)
- 'U' -- from (X, Y) moves to point (X - 1, Y)
- 'D' -- from (X, Y) moves to point (X + 1, Y)

The weather is cold because it's winter now. Initially, all points are snowy. But if Benny have already visited some point at any time this point becomes icy (because Benny has just taken a shower). Every time, when Benny makes a step into icy points she slips and falls down.

You are given a string **S** which denotes a set of instructions for Benny. Your task is to calculate how may times she will fall down.

https://www.hackerearth.com/practice/algorithms/sorting/quick-sort/practice-problems/algorithm/benny-and-gifts-marcheasy-3/

Program :-

2. Earth and The Meteorites

Once upon a time, the Earth was a flat rectangular landmass. And there was no life. It was then that the sky lit up with meteorites falling from out of space. Wherever they fell on the planet, a river was born, which flowed in all 4 directions (North, East, West, South), till the waters reached the edge of the Earth and simply fell off into space.

Now, these rivers criss-crossed and divided the one huge landmass (Pangaea) into many smaller landmasses. Now the lifeless (there was no life, remember?), want to know the number of landmasses on the planet after all the meteorites have fallen. They also want to know the area of the smallest and largest landmass. Can you help the lifeless in this question?

https://www.hackerearth.com/practice/algorithms/sorting/quick-sort/practice-problems/algorithm/earthandthemeteorites-qualifier2/

Program :-

Skill Session:

1.Sam Height <HSBC>

Sam among his friends wants to go to watch a movie in Armstord Cinema.
There is something special about Armstord cinema whenever people come in the group here. They will get seats accordingly their heights. Sam as a curious guy always wants to sit in the middle as cinema has the best view from the middle.
Now, Sam as the leader of his group decides who will join him for the movie.
Initially, he has N−1 friends with him (N including him).
You are given N−1 numbers that represent the heights of Sam's friends.
You are given the height of Sam as well.
Now, Sam can do two operations:
1. He can call a new friend of height H.
2. He can cancel any of his friend invitations.

https://www.hackerearth.com/practice/algorithms/sorting/merge-sort/practice-problems/algorithm/alice-and-marks-hsbc-b18d5d01/

Program :-

2. Median Game

You are given an array A of N integers. You perform this operation N−2 times: For each contiguous sub array of **odd size** greater than 2, you find the median of each sub array (Say medians obtained in a move are M1, M2, M3,…, Mk). In each move, you remove the first occurrence of value min (M1, M2, M3,…, Mk) from the original array. After removing the element the array size reduces by 1 and no void spaces are left. For example, if we remove element 2 from the array {1, 2, 3, 4}, the new array will be {1, 3, 4}.
Print a single integer denoting the sum of numbers that are left in the array after performing the operations. You need to do this for T test cases.

https://www.hackerearth.com/practice/algorithms/sorting/merge-sort/practice-problems/algorithm/median-game-june-easy-19-3722be60/

Program :-

3. Friendly Neighbours

You have n non-empty sets of positive integers S1,S2,…,Sn. The i-th person can live in any house from the set Si. You have to choose a house for each person. More formally, you have to create an array A1,A2,…,An such that for all i, Ai∈Si and Ai denotes the house of the i-th person. Let Bi denote the distance between i-th person and the closest neighbor to his left (some person j≠i such that Aj<Ai and Aj is maximum). If he doesn't have any such neighbor, we say that Bi=0. Let Ci equivalently denote the distance to the closest neighbor to his right.

You would like to create A1,A2,…,An in such a way that ∑B+∑C is minimized.

Find and print the minimum possible value of ∑B+∑C.

https://www.hackerearth.com/practice/algorithms/sorting/merge-sort/practice-problems/algorithm/choose-one-c4672347/

Program :-

4. Different queries

You are given an array A of size N. You have to perform Q queries on the array. Each of these queries belongs to the following types of queries:

1. L R X : Add X to all elements in the range [L,R]
2. L R X : Set the value of all elements in the range [L,R] to X

However, it is not mandatory to perform the queries in order. Your task is to determine the lexicographically largest array that can be obtained by performing all the Q queries.

https://www.hackerearth.com/practice/algorithms/sorting/merge-sort/practice-problems/algorithm/jumbled-queries-afb23321/

Program :-

6. Let's swap:

Y is alone too and has a permutation p1,p2,...,pn of numbers from 1 to n.
Y thinks that a permutation p1,p2,...,pn beautifulness is defined as value of $\sum|pi-i|, 1 \leq i \leq n$.
Y can swap two elements of the permutation **at most once**, what is the maximum beautifulness that Y can get?

https://www.hackerearth.com/practice/algorithms/sorting/merge-sort/practice-problems/algorithm/lets-swap-5075ade8/

Program :-

7. Specialty of a sequence:

You are given a sequence A of length n and a number k. A number A[l] is special if there exists a contiguous sub array that contains exactly k numbers that are strictly greater than A[l]. The specialty of a sequence is the sum of special numbers that are available in the sequence. Your task is to determine the specialty of the provided sequence.

https://www.hackerearth.com/practice/algorithms/sorting/quick-sort/practice-problems/algorithm/lex-finds-beauty-0d0bc1b6/

Program :-

(For Evaluator's use only)

Comment of the Evaluator (if Any)	Evaluator"s Observation Marks Secured:_____out of _____ Full Name of the Evaluator: Signature of the Evaluator Date of Evaluation:

Week-4

Lab Session:

Date of the Session: ___/___/___ **Time of the Session: ____to____**

Prerequisite:

1) Structure definition for a node having next and previous pointers
2) Collection of node connected in both directions
3) Single linked list concepts
4) Dynamic Memory Allocation concepts

Pre Lab Task

1. In a browsing center, the customer Kishore is sitting in a computer and browse the face book. Once he opened the face book, he wants to have a page of his friend Hari and gave a friend request and come back to his home page and open his other friend Lalithas page and read all her status and going on doing like this. Implement this as an operation of creation, insertion, deletion and searching of friends using Double Linked List.

Sample Input/Output:

****Menu****
5) Creation
6) Display the Elements
7) Insertion at begin
8) Insertion at any position
9) Delete a node any position
10) Search a node in DLL
Enter your choice … 1 8
23 34 12 78 55 29 66 24
****Menu****
1) Creation
2) Display the Elements
3) Insertion at begin
4) Insertion at any position
5) Delete a node any position
6) Search a node in DLL
7) Enter your choice … 2
23 <--->34<--->12<--->78<--->55<--->29<--->66<--->24
****Menu****
1) Creation
2) Display the Elements
3) Insertion at begin
4) Insertion at any position
5) Delete a node any position
6) Search a node in DLL
Enter your choice … 6
Enter the friend ID to be searched! 66
Yes, Found.

Implement this program for all menu items and check it.

Program :-

In Lab Task

1. Inserting a Node Into a Sorted Doubly Linked List

Given a reference to the head of a doubly-linked list and an integer, data, create a new DoublyLinkedListNode object having data value data and insert it at the proper location to maintain the sort.

Example

Head refers to the list 1 <-> 2 <-> 4 -> NULL
Data = 3
Return a reference to the new list: 1 <-> 2 <-> 3 <-> 4 -> NULL

Function Description

Complete the sortedInsert function in the editor below.
sortedInsert has two parameters:
DoublyLinkedListNode pointer head: a reference to the head of a doubly-linked list
int data: An integer denoting the value of the data field for the DoublyLinkedListNode you must insert into the list.

Returns

DoublyLinkedListNode pointer: a reference to the head of the list.

https://www.hackerrank.com/challenges/insert-a-node-into-a-sorted-doubly-linked-list/problem?h_r=internal-search

Program :-

2. Delete duplicate-value nodes from a sorted linked list

You are given the pointer to the head node of a sorted linked list, where the data in the nodes is in ascending order. Delete nodes and return a sorted list with each distinct value in the original list. The given head pointer may be null indicating that the list is empty.

Example

head refers to the first node in the list 1 ->2 -> 2 -> 3 -> 3 -> 3 -> 3 -> NULL.

Remove 1 of the 2 data values and return head pointing to the revised list 1 -> 2 -> 3 -> NULL .

Function Description

Complete the removeDuplicates function in the editor below.

removeDuplicates has the following parameter:

SinglyLinkedListNode pointer head: a reference to the head of the list

https://www.hackerrank.com/challenges/delete-duplicate-value-nodes-from-a-sorted-linked-list/problem?h_r=internal-search

Program :-

3. Find the middle of a given linked list using recursion

Given a singly linked list, find middle of the linked list. For example, if given linked list is 1->2->3->4->5 then output should be 3.

If there are even nodes, then there would be two middle nodes, we need to print second middle element. For example, if given linked list is 1->2->3->4->5->6 then output should be 4.
Note: Insert function should add nodes in the linked list.

https://www.hackerearth.com/problem/algorithm/find-the-middle-of-a-given-linked-list-using-recursion/

Program :-

Post Lab Task

1. In a Coco game, Krishna, Vasu, Shiva, Ganesh, Sathesh, Naveen and Anand are playing at an University ground. The rally tag is given to Anand. Anand stood and ran around the players and push Krishna as his tag. Imitate it as a circular linked list of Single and Double direction list and display all the players name in the list. Also Find how is the shortest person in the list.

Input/Output Format:

*** Menu ***
1) Create the list
2) Display the list
3) Print the shortest person In the list.
4) Exit. Enter your choice: 1 7
Enter the name and height of the Players:
Krishna 5.2
Vasu 5.1
Shiva 5.3
Ganesh 5.4
Sathesh 4.9
Naveen 5.5
Anand 5.0
*** Menu ***
1) Create the list
2) Display the list
3) Print the shortest person In the list.
4) Exit.
Enter your choice: 2
The Players are:

Name	Height
Krishna	5.2
Vasu	5.1
Shiva	5.3
Ganesh	5.4
Sathesh	4.9
Naveen	5.5
Anand	5.0

*** Menu ***
1) Create the list
2) Display the list
3) Print the shortest person In the list.
4) Exit. Enter your choice: 3
The shortest player is Sathesh having 4.9 inches height.

Skill Session:

1. Print the Elements of a Linked List

Given a pointer to the head node of a linked list, print each node's data element, one per line. If the head pointer is null (indicating the list is empty), there is nothing to print.

Function Description

Complete the printLinkedList function in the editor below.

PrintLinkedList has the following parameter(s):

SinglyLinkedListNode head: a reference to the head of the list

https://www.hackerrank.com/challenges/print-the-elements-of-a-linked-list/problem?h_r=internal-search

Program :-

2. Linked list

You have been given a singly linked list of integers along with an integer 'N'. Write a function to append the last 'N' nodes towards the front of the singly linked list and returns the new head to the list.

https://www.codechef.com/problems/REC_05

Program :-

3. Insert a Node at the Tail of a Linked List

You are given the pointer to the head node of a linked list and an integer to add to the list. Create a new node with the given integer. Insert this node at the tail of the linked list and return the head node of the linked list formed after inserting this new node. The given head pointer may be null, meaning that the initial list is empty.

Function Description

Complete the insertNodeAtTail function in the editor below.

insertNodeAtTail has the following parameters:

-SinglyLinkedListNode pointer head: a reference to the head of a list

int data: the data value for the node to insert

https://www.hackerrank.com/challenges/insert-a-node-at-the-tail-of-a-linked-list/problem?h_r=internal-search

Program :-

4. Delete a Node

Delete the node at a given position in a linked list and return a reference to the head node. The head is at position 0. The list may be empty after you delete the node. In that case, return a null value.

Example

Llist = 0 -> 1 -> 2 -> 3

Position = 2

After removing the node at position 2, llist = 0 -> 1-> 3.

Function Description

Complete the deleteNode function in the editor below.

deleteNode has the following parameters:

-SinglyLinkedListNode pointer list: a reference to the head node in the list

-int position: the position of the node to remove

https://www.hackerrank.com/challenges/delete-a-node-from-a-linked-list/problem?h_r=internal-search

Program :-

5. Remove Kth Node

A linked list contains N nodes numbered from 1 to N. The tail of the list points to head of the list i.e. the linked list is circular in nature. For when N=5 1->2->3->4, ^-------5<---' An integer K is also given to you. You start counting from head and when you reach Kth node in the circular list you remove that node. For eg. if K=7 we start from head i.e. 1 right now nd move K steps ahead till we reach node numbered 3 in circular list 1->2->3->4->5->1->2->[3] <---- we remove this node now the new linked list looks like this 1->2->4, ^---5<---' Node numbered 3 has been removed and now we start counting from the next node. This process is repeated until only one Node is left. Your task is to determine the number of the last node.

https://www.hackerearth.com/problem/algorithm/remove-kth-node/

Program :-

5. Compare two linked lists

You're given the pointer to the head nodes of two linked lists. Compare the data in the nodes of the linked lists to check if they are equal. If all data attributes are equal and the lists are the same length, return 1. Otherwise, return 0.

Example

llist1: 1->2->3->NULL

llist2: 1---->2--à3---à4-àNULL

The two lists have equal data attributes for the first 3 nodes. Llist2 is longer, though, so the lists are not equal. Return 0.

Function Description

Compare_lists has the following parameters:

SinglyLinkedListNode llist1: a reference to the head of a list

SinglyLinkedListNode llist2: a reference to the head of a list

https://www.hackerrank.com/challenges/compare-two-linked-lists/problem?h_r=internal-search

Program :-

(For Evaluator's use only)

Comment of the Evaluator (if Any)	Evaluator"s Observation Marks Secured:_____out of _____ Full Name of the Evaluator: Signature of the Evaluator Date of Evaluation:

Week-5

Lab Session:

Date of the Session: ___/___/___ **Time of the Session: ____to____**

Prerequisite

1) Single linked list
2) Double linked list
3) Structure definition
4) Dynamic memory allocation
5) C-SLL, C-DLL

Pre Lab Task

1. In a railway station an engine is connected with the N number of Bogies which are connected in one direction from engine to the last bogie. Implement this as an operation of creation, insertion, deletion and searching of a bogie using Single Linked List.

Sample Input/Output:

```
****Menu****
 1) Creation
 2)  Display the Elements
 3)  Insertion at begin
 4)  Insertion at any position
 5) Delete a node any position
 6) Search a node in SLL
 7) Enter your choice …
  1 6
  23 34 12 78 55 29
****Menu****
  1) Creation
  2) Display the Elements
  3) Insertion at begin
  4) Insertion at any position
  5) Delete a node any position
  6) Search a node in SLL
  7) Enter your choice … 2
   The Nodes are: ->23 -> 34 -> 12 -> 78 -> 55 -> 29
  ****Menu****
   1) Creation
   2) Display the Elements
   3) Insertion at begin
   4) Insertion at any position
   5) Delete a node any position
   6) Search a node in SLL
   Enter your choice … 5
   The node value 55 is deleted
   ****Menu****
    1) Creation
    2) Display the Elements
```

3) Insertion at begin
4) Insertion at any position
5) Delete a node any position
6) Search a node in SLL
Enter your choice … 2
The Nodes are: ->23 -> 34 -> 12 -> 78 -> 29
Test the above program using SLL for all menu options.

Program :-

In Lab Task

1. Reverse a List

You are given a list of elements. Reverse the list without using the reverse function. The input and output portions will be handled automatically. You need to write a function with the recommended method signature.

https://www.hackerrank.com/challenges/fp-reverse-a-list/problem?h_r=internal-search

Program :-

2. Reverse a doubly linked list

Given the pointer to the head node of a doubly linked list, reverse the order of the nodes in place. That is, change the next and prev pointers of the nodes so that the direction of the list is reversed. Return a reference to the head node of the reversed list.

Note: The head node might be NULL to indicate that the list is empty.

Function Description

Complete the reverse function in the editor below.

reverse has the following parameter(s):
DoublyLinkedListNode head: a reference to the head of a DoublyLinkedList

Returns

- DoublyLinkedListNode: a reference to the head of the reversed list

https://www.hackerrank.com/challenges/one-month-preparation-kit-reverse-a-doubly-linked-list/problem?h_r=internal-search

Program :-

3. Circular Doubly Linked list

You are given a circular doubly linked list that contains integers as the data in each node. These data on each node is distinct. You have developed a special algorithm that prints the three continuous elements of the list, starting from the first element or head of the list and runs for infinite time. For example, if the list is {1,9,12,7}, then the output of the algorithm will be {1,9,12,9,12,7,12,1..}. The output contains the infinite number of elements because it is a circular list. You are given only a part of the output that has been returned by the algorithm. Your task is to determine the number of elements available in the original list and print the respective elements.

https://www.hackerearth.com/problem/algorithm/hiddent-doubly-linked-liste8c1fead/

Program :-

Post Lab Task

1. In a Coco game, Krishna, Vasu, Shiva, Ganesh, Satheesh, Naveen and Anand are playing at an University ground. The rally tag is given to Anand. Anand stood and ran around the players and push Krishna as his tag. Imitate it as a circular linked list of Single and Double direction list and display all the players name in the list. Also Find how is the shortest person in the list.

Input/Output Format:

*** Menu ***
1) Create the list
2) Display the list
3) Print the shortest person in the list.
4) Exit.
Enter your choice:
1
7
Enter the name and height of the Players:
Krishna 5.2
Vasu 5.1
Shiva 5.3
Ganesh 5.4
Satheesh 4.9
Naveen 5.5
Anand 5.0
*** Menu ***
1) Create the list
2) Display the list
3) Print the shortest person in the list.
4) Exit.
Enter your choice: 2
The Players are:
Name Height
Krishna 5.2
Vasu 5.1
Shiva 5.3
Ganesh 5.4
Satheesh 4.9
Naveen 5.5
Anand 5.0
*** Menu ***
1) Create the list
2) Display the list
3) Print the shortest person In the list.
4) Exit.
Enter your choice: 3
The shortest player is Satheesh having 4.9 inches height

Program :-

Skill Session

1. Remove Friends

After getting her PhD, Christie has become a celebrity at her university, and her facebook profile is full of friend requests. Being the nice girl she is, Christie has accepted all the requests.

Now Kuldeep is jealous of all the attention she is getting from other guys, so he asks her to delete some of the guys from her friend list.

To avoid a 'scene', Christie decides to remove some friends from her friend list, since she knows the popularity of each of the friend she has, she uses the following algorithm to delete a friend.

Algorithm Delete(Friend): DeleteFriend=false for i = 1 to Friend.length-1 if (Friend[i].popularity < Friend[i+1].popularity) delete i th friend DeleteFriend=true break if(DeleteFriend == false) delete the last friend

https://www.hackerearth.com/practice/data-structures/linked-list/singly-linked-list/practice-problems/algorithm/remove-friends-5/

Program :-

2. Cycle Detection

A linked list is said to contain a cycle if any node is visited more than once while traversing the list. Given a pointer to the head of a linked list, determine if it contains a cycle. If it does, return 1 . Otherwise, return 0.

Example

head refers to the list of nodes 1-> 2 -> 3 -> NULL

The numbers shown are the node numbers, not their data values. There is no cycle in this list so return 0.

head refers to the list of nodes 1 -> 2 -> 3 -> 1 -> NULL

There is a cycle where node 3 points back to node 1, so return .

Function Description

Complete the has_cycle function in the editor below.

It has the following parameter:

· SinglyLinkedListNode pointer head: a reference to the head of the list

Returns

- Int: 1 if there is a cycle or 0 if there is not

https://www.hackerrank.com/challenges/detect-whether-a-linked-list-contains-a-cycle/problem?

Program :-

3. Reverse a linked list

Given the pointer to the head node of a linked list, change the next pointers of the nodes so that their order is reversed. The head pointer given may be null meaning that the initial list is empty.
Example head references the list 1-> 2-> 3-> NULL
Manipulate the pointers of each node in place and return head, now referencing the head of the list
3 -> 2 -> 1-> NULL.

Function Description

Complete the reverse function in the editor below.
Reverse has the following parameter:
-SinglyLinkedListNode pointer head: a reference to the head of a list

Returns

- SinglyLinkedListNode pointer: a reference to the head of the reversed list

https://www.hackerrank.com/challenges/reverse-a-linked-list/problem?h_r=internal-search

Program :-

4. Merge two sorted linked list

You‟re given the pointer to the head nodes of two sorted linked lists. The data in both lists will be sorted in ascending order. Change the next pointers to obtain a single, merged linked list which also has data in ascending order. Either head pointer given may be null meaning that the corresponding list is empty.

https://www.hackerrank.com/challenges/merge-two-sorted-linkedlists/problem

Program :-

5. Insert a node at the head of a linked list

Given a pointer to the head of a linked list, insert a new node before the head. The next value in the new node should point to head and the data value should be replaced with a given value. Return a reference to the new head of the list. The head pointer given may be null meaning that the initial list is empty.

Function Description

Complete the function insertNodeAtHead in the editor below.

insertNodeAtHead has the following parameter(s):

· SinglyLinkedListNode llist: a reference to the head of a list

· data: the value to insert in the data field of the new node

https://www.hackerrank.com/challenges/insert-a-node-at-the-head-of-a-linked-list/problem?h_r=internal-search

Program :-

6. Get Node Value

Given a pointer to the head of a linked list and a specific position, determine the data value at that position. Count backwards from the tail node. The tail is at position 0, its parent is at 1 and so on.

Example refers to

Each of the data values matches its distance from the tail. The value is at the desired position.

Function Description

Complete the getNode function in the editor below.

getNode has the following parameters:

SinglyLinkedListNode pointer head: refers to the head of the list

int positionFromTail: the item to retrieve

Returns

int: the value at the desired position

https://www.hackerrank.com/challenges/get-the-value-of-the-node-at-a-specific-position-from-the-tail/problem?h_r=internal-search

Program :-

(For Evaluator's use onl

Comment of the Evaluator (if Any)	Evaluator"s Observation Marks Secured:_____out of _____ Full Name of the Evaluator: Signature of the Evaluator Date of Evaluation:

Week-6

Lab Session:

Date of the Session: ___/___/___ **Time of the Session: ____to____**

Prerequisite

1. Stack – First in last out
2. One opening for storing the data
3. Initially Top = -1
4. Operation – Insertion – Push()
5. Operation – Deletion – Pop()
6. Applications of stack

Pre Lab Task

1.You are given a stack of N integers. In one operation, you can either pop an element from the stack or push any popped element into the stack. You need to maximize the top element of the stack after performing exactly K operations. If the stack becomes empty after performing K operations and there is no other way for the stack to be non-empty, print -1.

Program :-

2. In this problem, element is divided by 2 and remainders are pushed into the stack until remainder becomes zero and finally remainders are popped out of the stack to get equivalent binary number.

Program :-

In Lab Task

1. Maximum Element

You have an empty sequence, and you will be given queries. Each query is one of these three types:

1. Push the element x into the stack.

2. Delete the element present at the top of the stack.

3. Print the maximum element in the stack.

https://www.hackerrank.com/challenges/maximum-element/problem?isFullScreen=true

Program :-

2. Equal Stacks

You have three stacks of cylinders where each cylinder has the same diameter, but they may vary in height. You can change the height of a stack by removing and discarding its topmost cylinder any number of times.

Find the maximum possible height of the stacks such that all of the stacks are exactly the same height. This means you must remove zero or more cylinders from the top of zero or more of the three stacks until they are all the same height, then return the height.

https://www.hackerrank.com/challenges/equal-stacks/problem?isFullScreen=true

Program :-

3. Waiter

you are a waiter at a party. There is a pile of numbered plates. Create an empty array. At each iteration, , remove each plate from the top of the stack in order. Determine if the number on the plate is evenly divisible by the prime number. If it is, stack it in pile . Otherwise, stack it in stack . Store the values in from top to bottom in . In the next iteration, do the same with the values in stack . Once the required number of iterations is complete, store the remaining values in in , again from top to bottom. Return the array.

https://www.hackerrank.com/challenges/waiter/problem?isFullScreen=true

Program :-

Post Lab Task

1. In common terminology, we use the stack with same meaning like stack of CDs or stack of DVDs. Extend the concept of stack in real life, you remove the CD at the top before moving CD below it. So the CD which goes last on to stack, come off the stack first. We can make use of stack to reverse a given string. To reverse a word. You push a given word to stack - letter by letter - and then pop letters from the stack.

Program:-

2. Implement a program in C to input a two-digit number and store all its divisors in a stack. The program should display divisors.

Solution:-

Skill Session:

1. Sudhanva and Books

xAs we all know, Sudhanva is fond of reading books and taking photographs. His dad (after gifting him an iPad last birthday) gifted him a custom-made bookshelf this birthday. This shelf is made in such a way that books can be inserted and removed only from the top. Books are placed one over the other in that shelf. Now to remember each book, Sudhanva develops an algorithm. He first writes a number in his iPad, then takes a photo of that number in his DSLR and then pastes that photograph on the cover of each book. Initially the bookshelf is empty.

You have given QQ queries of either of the two types

• Type 1. (1,N)(1,N) In this type of query Sudhanva places a book with photo of number N on the top. You don't have to print anything in this query

• Type 2. (−1)(−1) In this Query Sudhanva takes out the topmost book, you have to print the number on the book he just took out. If the shelf is empty and Sudhanva is trying to take out a book, print "kuchbhi?" Without quotes

https://www.hackerrank.com/challenges/waiter/problem?isFullScreen=true

Program :-

2. Simple stack

You have to find the deepest position of an element in the given stack of numbers.
First line of input will contain an integer N, denoting the number of elements in stack.
Following N lines will contain elements E starting from bottom and ending at top.
last line will contain an integer x, whose deepest position is to be found.
Your output should contain 1 line, printing the deepest index (1-based index) value of x
https://www.hackerearth.com/problem/algorithm/simple-stack/

Program :-

3. Stack operations

You are given a stack of N integers. In one operation, you can either pop an element from the stack or push any popped element into the stack. You need to maximize the top element of the stack after performing exactly K operations. If the stack becomes empty after performing K operations and there is no other way for the stack to be non-empty, print -1.

https://www.hackerearth.com/practice/data-structures/stacks/basics-of-stacks/practice-problems/algorithm/stakth-1-e6a76632/

Program :-

4. Stack using arrays

The first input is the number of test cases. Thereafter each test case starts with n, the number of inputs. Thereafter, each test case starts with n, the number of commands and s, the maximum stack size (the maximum number of elements that can be stored in the stack at any point of time). Thereafter, each of the following n lines have commands “push a” meaning push integer a in the stack; “pop” meaning pop an item from the stack and print on the screen; and “top”, meaning print the element at the top of the stack on the screen.

https://www.hackerearth.com/problem/algorithm/stack-using-arrays-2/

Program :-

5. Mayank and his stacks

Mayank has three stacks of cylinders where each cylinder has the same diameter, but they may vary in height. He can change the height of a stack by removing and discarding its topmost cylinder any number of times.

Find the maximum possible height of the stacks such that all of the stacks are exactly the same height. This means he must remove zero or more cylinders from the top of zero or more of the three stacks until they are all the same height, then return the height. Help Mayank !

https://www.hackerearth.com/problem/algorithm/mayank-and-his-stacks/

Program :-

6. Queues and Stacks

Palindrome is a word, phrase, number, or other sequence of characters which reads the same backwards and forwards. Can you determine if a given string, , is a palindrome?.To solve this challenge, we must first take each character in , enqueue it in a queue, and also push that same character onto a stack. Once that's done, we must dequeue the first character from the queue and pop the top character off the stack, then compare the two characters to see if they are the same; as long as the characters match, we continue dequeueing, popping, and comparing each character until our containers are empty (a non-match means isn't a palindrome).

https://www.hackerrank.com/challenges/30-queues-stacks/problem?h_r=internal-search

Program :-

(For Evaluator's use only)

Comment of the Evaluator (if Any)	Evaluator"s Observation Marks Secured:_____out of _____ Full Name of the Evaluator: Signature of the Evaluator Date of Evaluation:

Week-7

Lab Session:

Date of the Session: ___/___/___ **Time of the Session: ____to____**

Prerequisite

- Queue – First in First out
- Queue has two openings
- Initially FRONT = REAR = -1
- Operation – Insertion – Enqueue()
- Operation – Deletion – Dequeue()

Pre lab Task

1. Show the detailed contents of the stack after performing the following operations and write routines to implement push and pop operations. enqueue('P'), enqueue('U'), dequeue(), enqueue('S'), enqueue('H'), dequque(), enqueue('P'), dequeue(),enqueue('A').

2.. In a railway station an engine is connected with the N number of Bogies which are connected in one direction from engine to the last bogie. Implement queue using SLL. Operations are creation, insertion, deletion and searching of a bogie using Single Linked List. Test the above program using SLL for all menu options.

In Lab Task

1. Max in Queue

You have been given a sequence A of N digits. Each digit in this sequence ranges from 1 to 109 . You need to perform 2 types of operations on this list:
Add(x): Add element x to the end of the list.
Max(list): Find the maximum element in the current sequence.
For each query of type 2, you need to print the result of that operation.

https://www.hackerearth.com/practice/data-structures/trees/heapspriority-queues/practice-problems/algorithm/queues-content-problem/

Program :-

2. Code Queue

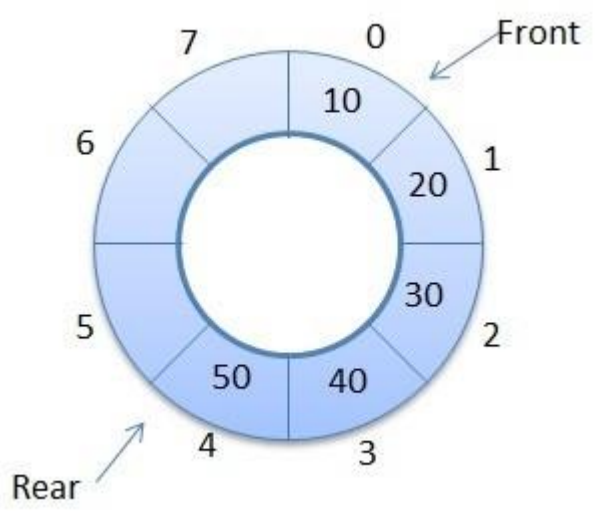
You are working with a high-class city mall development plan which is based on a full automation system. The owner has said to you that the entry for movies should be in such a way that all those who have registered for the movie can enter in easy whereas the one who hasn't registered shall not be allowed.
All technical work has been done and you as a programmer have to develop a queue program in order to meet the required conditions by the owner. You can perform all the queue operations. During the input, the customer name and tickets are to be uploaded and further in the program, the user shall enter its name and ticket number which is to be checked in the database and the required message is too displayed.

https://www.codechef.com/problems/KCPROG4

Program :-

3. Circular Queue using Arrays

Implement a Circular Queue using arrays with the following operations.
I**nsert** (Display "Queue Overflow" if queue is full).
Delete (Display "Queue Underflow" if queue is empty).
Display (Display NULL if queue is empty).
Exit
*Note: The circular queue is a linear data structure. It follows FIFO principle. In circular queue, the last node is connected back to the first node to make a circle.



https://www.hackerrank.com/contests/17cs1102/challenges/7a-circular-queue-using-arryas

Program :-

Post lab Tsk

1. Show the detailed contents of the queue data structure by inserting numbers 10, 15, 5, 25, 17, 21, 7, 30. What would be the contents of the queue after performing the following operations? i. Delete two elements. ii. Insert 37 and 20.

2. **Implement a queue using the stack data structure**

The idea is to implement the queue's enqueue operation so that the first entered element always ends up at the top of the stack. To achieve this, we need an additional stack. To enqueue an item into the queue, first move all elements from the first stack to the second stack, push the item into the first stack, and finally move all elements back to the first stack. This ensures that the new item lies at the bottom of the stack and hence would be the last one to be removed. To dequeue an item from the queue, return the top item from the first stack.

Solution:-

Skill Session:

1. Chefs in Queue

All the chefs (except the Head Chef) are standing in queue to submit their bills. The chefs have different seniority. In all there are **N** chefs of **K** different seniority levels. These are given to you in an array, where $\mathbf{A_1}$, $\mathbf{A_2}$, ..., $\mathbf{A_N}$ denote the seniority of chefs in the queue. $\mathbf{A_N}$ denotes front of the queue and $\mathbf{A_1}$ denotes end of the queue.
Head Chef gets an interesting thought past his head. He begins to think what if every chef starting from the end of the queue begins to delegate his job of submitting bills to a chef least ahead of him in the queue but junior to him. The Head Chef's fearfulness of this scenario is **f = i2 - i1 + 1**, where **i1** is the index of chef in queue and **i2** is the index of the junior chef. Head Chef's total fearfulness of chaos is the product of all the fearfulness in Head Chef's mind. Note if a chef has no junior ahead of him/her in the queue then Head Chef's fearfulness for this Chef is **1**.
You are required to find the Head Chef's total fearfulness given an arrangement of Chef's in a queue. Since this number can be quite large output it modulo **1000000007**.

https://www.codechef.com/problems/CHFQUEUE

Program :-

2. Dr Phil goes to the ranch

Dr. Phil has put you in charge of his new hospital while he is chilling at the ranch. NN people have booked appointment with the Doctor today numbered from 11 to NN such that person XX booked appointment before person YY if X<YX<Y.
Now, the people do not necessarily arrive at the hospital on time, so you decide of an unusual way of handling the situation. At the start you fulfill the appointment of the people who booked first (so if person 1 is available then that person can meet the doctor) otherwise you look for next available person.
If you fulfill the appointment of any person and their exist some people (1 or more) who have booked earlier but haven't arrived at the hospital yet, then those people are said to have missed their appointment.
If at some point of time there exists a set of people that have missed their appointment earlier and the person who booked appointment most recently among them (later than others in that set) say person XX is now available then XX is given the priority to meet the doctor over any other person. But even if any person in that set YY other than XX is available and XX is not available then you start appointments of the people who have not yet missed their appointments in their order of booking.
You are given the order in which people arrive at the hospital and have to print the order of their appointments.
Note: Use fast input/output methods. Python users should submit their solutions in PYPY 3.

https://www.codechef.com/problems/CAC202

Program :-

3. Queue Problem

Students are waiting in the queue to buy the movie ticket. You have to find the number of arrangements of queue such that no. of boys is always greater than no. of girls at any position.
For example: BBBGG, BBGBG are valid arrangements while BGBBG, BBGGB are invalid arrangements in which B denotes a Boy and G denotes a Girl.
Note that every person is considered as a distinct person. Therefore, B1B2G1B3G2 and B1B3G1B2G2 are considered as distinct arrangements.

https://www.hackerearth.com/problem/algorithm/queue-problem-jatinj-1addbbb7/

Program :-

4. Aniruddha's Queue

Aniruddha is given a milestone M to reach in terms of distance. He is living in a different Galaxy where there are N days in a year.At the ith day he can walk atmost X distance.Assuming he walks optimally you need to output the minimum day number on which he will reach the milestone.

https://www.hackerearth.com/practice/basic-programming/implementation/basics-of-implementation/practice-problems/algorithm/aniruddhas-queue-4/

Program :-

5. queue-using-two-stacks

A queue is an abstract data type that maintains the order in which elements were added to it, allowing the oldest elements to be removed from the front and new elements to be added to the rear. This is called a *First-In-First-Out* (FIFO) data structure because the first element added to the queue (i.e., the one that has been waiting the longest) is always the first one to be removed.
A basic queue has the following operations:
Enqueue: add a new element to the end of the queue.
Dequeue: remove the element from the front of the queue and return it.
In this challenge, you must first implement a queue using *two stacks*. Then process queries, where each query is one of the following types:
1 x: Enqueue element into the end of the queue.
2: Dequeue the element at the front of the queue.
3: Print the element at the front of the queue.

https://www.hackerrank.com/challenges/queue-using-two-stacks/problem?isFullScreen=true

Program :-

6. Long ATM Queue

Due to the demonetization move, there is a long queue of people in front of ATMs. Due to withdrawal limit per person per day, people come in groups to withdraw money. Groups come one by one and line up behind the already present queue. The groups have a strange way of arranging themselves. In a particular group, the group members arrange themselves in increasing order of their height(not necessarily strictly increasing).
Swapy observes a long queue standing in front of the ATM near his house. Being a curious kid, he wants to count the total number of groups present in the queue waiting to withdraw money. Since groups are standing behind each other, one cannot differentiate between different groups and the exact count cannot be given. Can you tell him the minimum number of groups that can be observed in the queue?
The first line of input contains one positive integer N. The second line contains N space-separated integers H[i] denoting the height of i-th person. Each group has group members standing in increasing order of their height.

https://www.hackerearth.com/practice/data-structures/arrays/1-d/practice-problems/algorithm/long-atm-queue-3/

Program :-

(For Evaluator's use only)

Comment of the Evaluator (if Any)	Evaluator"s Observation Marks Secured:_____out of _____ Full Name of the Evaluator: Signature of the Evaluator Date of Evaluation:

Week-8

Lab Session:

Date of the Session: ___/___/___ **Time of the Session: ____to____**

Prerequisite:

- Singly linked List
- Stack operations

Pre Lab Task

1. Convert an infix expression into postfix expression.

Note: In case two operators have equal precedence, the order of operation goes from left to right

Input Description

The first input will be a single integer N denoting the number of test cases to take. After this there will be exactly N lines, each line a valid infix string. The string will be a valid postfix expression consisting of integers, binary operators (+, -, *, / and $) and parenthesis. Every integer, operator and parenthesis will be compulsorily separated by a SPACE. The symbol „?" denotes the end of expression.

Output Description

Exactly N lines, each line denoting the postfix expression. Every integer and operator must be separated by a single SPACE.

Sample Input 1 31 * (4 + 50)

Sample Output 31 4 50 + *

Sample Output ABC*DEF^/G*-H*+

Program :-

2. Evaluate Postfix Expression

Input Description

The first input will be a single integer N denoting the number of test cases to take. After this there will be exactly N lines, each line a valid postfix string. The string will be a valid postfix expression consisting of only integers and binary operators (+, -, *, / and $). Every integer and operator will be compulsorily separated by a SPACE. The symbol „?" denotes the end of expression.

Output Description

Exactly N lines, each line denoting the output of the expression. In case the output is in fractions, please print only the integer part of the output.

Sample Input 1 31 4 50 + * ?

Sample Output 1674

Program :-

In Lab Task

1. **Postfix Expression Evaluation**
Implement a program to evaluate a postfix expression.
Input Format
The first input will be a single integer N denoting the number of test cases to take. After this there will be exactly N lines, each line a valid postfix string. The string will be a valid postfix expression consisting of only integers and binary operators (+, -, *, / and ?).Every integer and operator will be compulsorily separated by a SPACE. The symbol '?' denotes the end of expression.

https://www.hackerrank.com/contests/17cs1102/challenges/8-c-postfix-expression-evaluation/problem

Program :-

2. Infix to Postfix

My Laptop, Mac, really hates every time I write an infix expression! You have to ease the task for Mac by transforming a given Infix expression to Postfix. Instead of numeric variables the expression given is using lowercase english letters (a to z). The priority of operators is as follows '+', '-', '*', '/', '^' (increasing left to right) '^' is for exponentiation. Each case would be a string without space in valid form. There won't be any case with ambiguity i.e., such that two different answers are possible e.g., (a+b+c), it would be well parenthesised in that case.

https://www.codechef.com/problems/INFPOS03

Program :-

3. Check for balanced parentheses in an expression

Problem Description
you have been given an expression containing parenthesis that you will be provided as input you have to check whether the expression has balanced parenthesis or not.

Input Format
input will be in form of brackets.

Constraints
your code must be memory efficient

Output Format
output must be : 0 for not-balanced 1 for balanced

Sample Input 0

{(())}
Sample Output 0
1
Sample Input 1
{()
Sample Output 1
0

https://www.hackerrank.com/contests/the-great-programming-challange/challenges/check-for-balanced-parentheses-in-an-expression

Program :-

Post Lab Task:

1. Draw the expression tree and find the infix and prefix expressions for the following postfix expression: AB*CD/+EF-*.

Solution:-

2. Display the postfix expression corresponding to the infix expression I-J/K+L ^ M ^ N. Assume that the operators +, -, / are left associative and ^ is right associative. The order of precedence (from highest to lowest) is ^, /, +, - and evaluate it for the given data I=2, J=4, K=1, L=3, M=-2, N=5.

Solution:-

Skill Session:

1. Transform the expression

Transform the given infix expression to postfix form

https://www.hackerearth.com/problem/algorithm/transform-the-expression-2/

Program :-

2. Check for balanced parentheses in an expression

Problem Description

you have been given an expression containing parenthesis that you will be provided as input you have to check whether the expression has balanced parenthesis or not.

https://www.hackerrank.com/contests/the-great-programming-challange/challenges/check-for-balanced-parentheses-in-an-expression

Program :-

3. Balanced Brackets

A bracket is considered to be any one of the following characters: (,), {, }, [, or].
Two brackets are considered to be a matched pair if the an opening bracket (i.e., (, [, or {) occurs to the left of a closing bracket (i.e.,),], or }) of the exact same type. There are three types of matched pairs of brackets: [], {}, and ().
A matching pair of brackets is not balanced if the set of brackets it encloses are not matched. For example, {[(])} is not balanced because the contents in between { and } are not balanced. The pair of square brackets encloses a single, unbalanced opening bracket, (, and the pair of parentheses encloses a single, unbalanced closing square bracket,].
By this logic, we say a sequence of brackets is balanced if the following conditions are met:
It contains no unmatched brackets.
The subset of brackets enclosed within the confines of a matched pair of brackets is also a matched pair of brackets.
Given strings of brackets, determine whether each sequence of brackets is balanced. If a string is balanced, return YES. Otherwise, return NO.

Function Description

Complete the function isBalanced in the editor below.
isBalanced has the following parameter(s):
string s: a string of brackets

Returns

string: either YES or NO

Program:-

4. **Evaluate Expression**

Problem description

You are given fully parenthesized expression. You have to evaluate the expression and give the result.

https://www.hackerrank.com/contests/cs1300-odd-2014/challenges/evaluate-expression/problem

Program:-

5. **Stack and Queue <Nissan>**

You are given a stack of N integers such that the first element represents the top of the stack and the last element represents the bottom of the stack. You need to pop at least one element from the stack. At any one moment, you can convert stack into a queue. The bottom of the stack represents the front of the queue. You cannot convert the queue back into a stack. Your task is to remove exactly K elements such that the sum of the K removed elements is maximised.

Input format :

The first line consists of two space-separated integers N and K.

The second line consists of N space-separated integers denoting the elements of the stack.

Output format :

Print the maximum possible sum of the K removed elements

Constraints :

$1 \leq N \leq 105$

$1 \leq K \leq N$

$1 \leq Ai \leq 109$

Sample Input

10 5

10 9 1 2 3 4 5 6 7 8

Sample Output

40

Explanation

Pop two elements from the stack. i.e {10, 9}

Then convert the stack into queue and remove first three elements from the queue. i.e {8, 7, 6}.

The maximum possible sum is 10+9+8+7+6 = 40

Program:-

6. Disk tower

Your task is to construct a tower in N days by following these conditions:
Every day you are provided with one disk of distinct size.
The disk with larger sizes should be placed at the bottom of the tower.
The disk with smaller sizes should be placed at the top of the tower.
The order in which tower must be constructed is as follows:
You cannot put a new disk on the top of the tower until all the larger disks that are given to you get placed.
Print N lines denoting the disk sizes that can be put on the tower on the ith day.

https://www.hackerearth.com/practice/data-structures/queues/basics-of-queues/practice-problems/algorithm/disk-tower-b7cc7a50/

Program:-

7. Infix to Postfix

My Laptop, Mac, really hates every time I write an infix expression! You have to ease the task for Mac by transforming a given Infix expression to Postfix. Instead of numeric variables the expression given is using lowercase english letters (a to z). The priority of operators is as follows '+', '-', '*', '/', '^' (increasing left to right) '^' is for exponentiation. Each case would be a string without space in valid form. There won't be any case with ambiguity i.e., such that two different answers are possible e.g., (a+b+c), it would be well parenthesized in that case.
https://www.codechef.com/problems/INFPOS03

Program:-

Comment of the Evaluator (if Any)	Evaluator‟s Observation Marks Secured:_____out of _____ Full Name of the Evaluator: Signature of the Evaluator Date of Evaluation:

Week-9

Lab Session:

Date of the Session: ___/___/___ **Time of the Session: ____to____**

Prerequisite:

- Hashing, hashing function, Hashing table
- Node – structure definition for Hashing
- Dynamic memory allocation
- Types of hashing

Pre-Lab Task:

1. Construct a hash table for the given input sequence: 461, 137, 675, 197, 294, 965, 131 and hash function H(k) = k mod 10. Use separate chaining if collision occurs.

Solution:-

2. The keys 12, 18, 13, 2, 3, 23, 5 and 15 are initially inserted into an empty hash table of size 11 using open addressing with hash function h(k) = k mod 11 and linear probing. Show the resultant hash table?

Solution:-

In-Lab Task:

1. Unusual construction

There are *N* cities in a country. There are *M* roads with parameters *(L, R, W)* denoting cities *L* and *R* that are connected and a road that was built between them for *W$*. Note that the roads are bidirectional in nature.
Now, a new policy is introduced to act that states the following:

Let *X* be the maximum number of roads between any pair of cities. All the pairs of cities that have fewer than *X* roads between them will be reconstructed and the cost of building the roads will be the same as before.

Determine the amount of money that will be required for implementing the new policy?

https://www.hackerearth.com/practice/data-structures/hash-tables/basics-of-hash-tables/practice-problems/algorithm/unusual-construction-3ec2e03f/

Program:-

2. Count Pairs

You are given an array *A* consisting of *N* non-negative integers. You are also given 2 integers $p(a \text{ prime number})$ and k. You are required to count number of pairs (i,j) where, $1 \leq i < j \leq N$ and satisying:

$$(A_i^2 + A_j^2 + A_i * A_j) \, mod \, p = k$$

where $a \, mod \, p = b$ means that b is the remainder when a is divided by p. In particular, $0 \leq b < p$.

https://www.hackerearth.com/practice/data-structures/hash-tables/basics-of-hash-tables/practice-problems/algorithm/count-pairs-9-d69fcdc3/

Program:-

3. Pairs of elements

You are given an array of length *N*. You are required to count the number of (i,j) pairs where $1 \leqslant i < j \leqslant N$ such that the difference of the array elements on that indices is equal to the sum of the square of their indices. That is the count of the number of pairs of (i,j) such that it satisfies this equation $(A[j] - A[i] = i^2 + j^2)$.

https://www.hackerearth.com/practice/data-structures/hash-tables/basics-of-hash-tables/practice-problems/algorithm/t-rex-and-the-pairs-0a045ce2/

Program:-

Post-Lab Task:

1. Insert the sequence of keys 43, 92, 123, 101, 36, 81, 52, 28 into hash table. Consider the hash table size 11 and hash function H(K) = K mod 11 and use quadratic probing technique for collision resolution?

Solution:-

2. Given the input {92, 58, 27, 35, 19, 79, 48, 64}, a fixed table size of 11, and a hash function H(X) = X mod 11, show the resultant hash table using double hashing, if collision occurs?

Solution:-

Skill Session:

1. **Plot the Curve**

You are given with integers $a, b, c, d, m.$ These represent the modular equation of a curve $y^2 mod\ m = (ax^3 + bx^2 + cx + d)\ mod\ m.$

Also, you are provided with an array A of size N. Now, your task is to find the number of pairs in the array that satisfy the given modular equation. If (A_i, A_j) is a pair then $A_j^2 mod\ m = (aA_i^3 + bA_i^2 + cA_i + d)\ mod\ m.$

Since the answer could be very large output it modulo $10^9 + 7.$

Note: A pair is counted different from some other pair if either A_i of the two pairs is different or A_j of the two pairs is different. Also for the convenience of calculations, we may count (A_i, A_i) as a valid pair if it satisfies given constraints.

https://www.hackerearth.com/practice/data-structures/hash-tables/basics-of-hash-tables/practice-problems/algorithm/lets-plot-this-47a575ed/

Program:-

2. Similar Chocolates

There are N chocolates denoted by array A where $A[i]$ is the length of the i^{th} chocolate. Alice can melt each chocolate and then convert it into a chocolate whose length is any divisor of the number $A[i]$. So, a chocolate of length $A[i]$ can be converted into X different types of chocolate where X is the count of divisors of the number $A[i]$. So you need to count the total unordered pair of chocolates such that their X value is same.

https://www.hackerearth.com/practice/data-structures/hash-tables/basics-of-hash-tables/practice-problems/algorithm/notebook-pages-dbad75a5/

Program:-

3.Maximum Sum

Given an array A of N integers. Now, you have to output the sum of unique values of the maximum sub array sum of all the possible sub arrays of the given array A.
Note: Sub array means contiguous elements with at least one element in it.

https://www.hackerearth.com/practice/data-structures/hash-tables/basics-of-hash-tables/practice-problems/algorithm/maximum-subarray-sum-of-subarrays-7f33aefa/

Program:-

4. **Festivals**

Alice likes festivals a lot as we all do. He also likes spending money on these festivals. He spends money in buying various things on these festivals. But he has problem of forgetting. He only remembers his top three maximum spending for any festival.
For e.g., on Holi he spends 25 units on colors, 50 units on water sprays, 100 units on gifts, 150 units on sweets but he remembers only his top 3 spending ,i.e., 50, 100 & 150.
Now as the year ends he wants to know the festival on which he spent most of his money that he can remember.

https://www.hackerearth.com/practice/data-structures/hash-tables/basics-of-hash-tables/practice-problems/algorithm/suzakus-festivals-14dacd7c/

Program:-

5. Bob and String

Bob and Khatu both love the string. Bob has a string **S** and Khatu has a string **T**. They want to make both string **S** and **T** to anagrams of each other. Khatu can apply two operations to convert string **T** to anagram of string **S** which are given below:

1) Delete one character from the string **T**.

2) Add one character from the string **S**.

Khatu can apply above both operation as many times he want. Find the minimum number of operations required to convert string **T** so that both **T** and **S** will become anagram of each other.

https://www.hackerearth.com/practice/data-structures/hash-tables/basics-of-hash-tables/practice-problems/algorithm/bob-and-string-easy/

Program:-

6. **ICPC Team Management.**

Little Chandan is an exceptional manager - apart from his role in HackerEarth - as the person who has to bug everyone, in general... and if possible, try to get some work done.

He's also offered a job as the coach of the best Russian teams participating for ACM-ICPC World Finals. Now, Chandan is an extremely good coach, too. But he's a weird person who thrives on patterns in life, in general. So, he has decided that if there are *n* number of students in total, and he is supposed to divide them in camps of *k* students - he want them to be arranged in such a way that the length of names of all the students in a camp is **equal.**

I know, totally weird, right?

https://www.hackerearth.com/practice/data-structures/hash-tables/basics-of-hash-tables/practice-problems/algorithm/icpc-team-management/

Program:-

(For Evaluator's use only)

Comment of the Evaluator (if Any)	Evaluator"s Observation Marks Secured:_____out of _____ Full Name of the Evaluator: Signature of the Evaluator Date of Evaluation:

Week-10

Lab Session:

Date of the Session: ___/___/___ **Time of the Session: ____to____**

Prerequisite:

- Tree, Sub-tree, Parent, Child - nodes
- Node – Degree, Height, Level, Depth
- Structure definition for a tree node
- Dynamic memory allocation

Pre-Lab Task:

A Binary Search Tree (BST) is a tree in which all the nodes follow the below-mentioned properties −

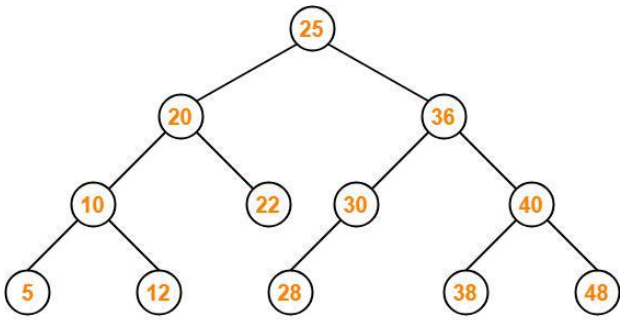
- The left sub-tree of a node has a key less than or equal to its parent node's key.
- The right sub-tree of a node has a key greater than to its parent node's key.

Thus, BST divides all its sub-trees into two segments; the left sub-tree and the right sub-tree and can be defined as −

left_subtree (keys) ≤ node (key) ≤ right_subtree (keys)

Representation: BST is a collection of nodes arranged in a way where they maintain BST properties. Each node has a key and an associated value. While searching, the desired key is compared to the keys in BST and if found, the associated value is retrieved.

1. For the given tree structure, traverse the in-order, pre order and post order



Binary Search Tree

In order:

Pre order:

Post order:

2. For the given node values, construct the BST structure and traverse in-order, pre order and post order. 25, 20, 35, 22, 12, 56, 18, 30, 38, 8, 44, 28, 10, 26, 5.

Solution:-

In-Lab Task

Problem Statement:

Complete the function in the editor(provided through link), which has parameter: a pointer to the root of a binary tree. It must print the values in the tree's pre-order traversal as a single line of space-separated values.

https://www.hackerrank.com/challenges/one-week-preparation-kit-tree-preorder-traversal/problem?h_r=internal-search

Program:-

2. **Binary Search Tree : Insertion**

Problem:

You are given a pointer to the root of a binary search tree and values to be inserted into the tree. Insert the values into their appropriate position in the binary search tree and return the root of the updated binary tree. You just have to complete the function.

https://www.hackerrank.com/challenges/binary-search-tree-insertion/problem

Program:-

3. **Binary Search Trees**

Problem Statement

The height of a binary search tree is the number of edges between the tree's root and its furthest leaf. You are given a pointer, root ,pointing to the root of a binary search tree. Complete the *getHeight* function provided in your editor so that it returns the height of the binary search tree.
https://www.hackerrank.com/challenges/30-binary-search-trees/problem

Program:-

Post-Lab Task:

1. Draw a complete binary tree with exactly six nodes. Put a different value in each node. Then draw an array with six components and show where each of the six node values would be placed in the array (using the usual array representation of a complete binary tree).

Solution:-

2. Given a binary search tree, we want to write a search method. This method will look for a node with a specific key and return that node. Below is an incomplete implementation

```
struct Node* search(Node, key):
{
        if(Node.data == Key )
                Return Node;
        else if(key>Node.data)
                ----------------------
        else
                ----------------------
}
```

The two lines empty indicate missing code. What lines should replace line 1 and line 2 respectively?

Solution:-

Skill Session:

1. *Transform the Expression*

Reverse Polish Notation (RPN) is a mathematical notation where every operator follows all of its operands. For instance, to add three and four, one would write "3 4 +" rather than "3 + 4". If there are multiple operations, the operator is given immediately after its second operand; so the expression written "3 − 4 + 5" would be written "3 4 − 5 +" first subtract 4 from 3, then add 5 to that.

Transform the algebraic expression with brackets into RPN form.

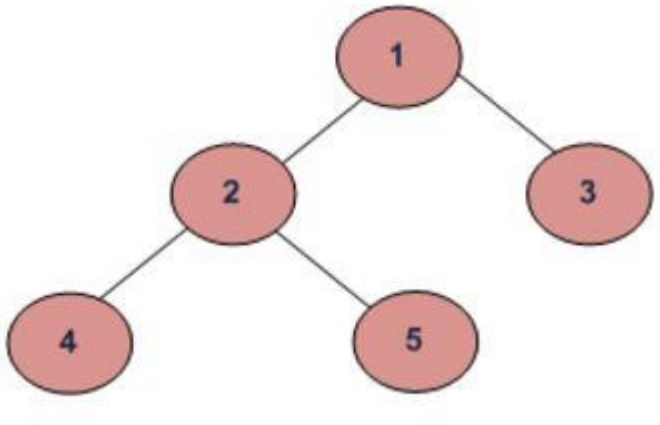
You can assume that for the test cases below only single letters will be used, brackets [] will not be used and each expression has only one RPN form (no expressions like a*b*c)

https://www.codechef.com/problems/ONP

Program:-

2. Count Number of Leaf Nodes in a tree

A node is a leaf node if both left and right child nodes of it are NULL.



Leaf count for the above tree is 3.

https://www.hackerearth.com/problem/algorithm/count-leaf-nodes-in-a-binary-tree/

Program:-

3. **Tree: Inorder Traversal**

In this challenge, you are required to implement inorder traversal of a tree.
Complete the function in your editor below, which has parameter: a pointer to the root of a binary tree.
It must print the values in the tree's inorder traversal as a single line of space-separated values.
https://www.hackerrank.com/challenges/tree-inorder-traversal/problem?h_r=internal-search

Program:-

4. Monk and his Friends

Problem

Monk is standing at the door of his classroom. There are currently **N** students in the class, **i**'th student got $\mathbf{A_i}$ candies.
There are still **M** more students to come. At every instant, a student enters the class and wishes to be seated with a student who has **exactly** the same number of candies. For each student, Monk shouts YES if such a student is found, NO otherwise.

https://www.hackerearth.com/practice/data-structures/trees/binary-search-tree/practice-problems/algorithm/monk-and-his-friends/

Program:-

5. **Tree: Post-order Traversal**

Complete the function in the editor below(in link below). It received parameter: a pointer to the root of a binary tree. It must print the values in the tree's post-order traversal as a single line of space-separated values.

https://www.hackerrank.com/challenges/tree-postorder-traversal/problem

Program:-

6. Create BST

Problem

Create a Binary Search Tree from list *A* containing *N* elements. Insert elements in the same order as given. Print the pre-order traversal of the subtree with root node data equal to *Q* (inclusive of *Q*), separating each element by a space.

https://www.hackerearth.com/practice/data-structures/trees/binary-search-tree/practice-problems/algorithm/create-bst/

Program:-

(For Evaluator's use only)

Comment of the Evaluator (if Any)	Evaluator"s Observation Marks Secured:_____out of _____ Full Name of the Evaluator: Signature of the Evaluator Date of Evaluation:

Week-11

Date of the Session: ___/___/___ **Time of the Session: _____to_____**

Lab Session:

Prerequisite:

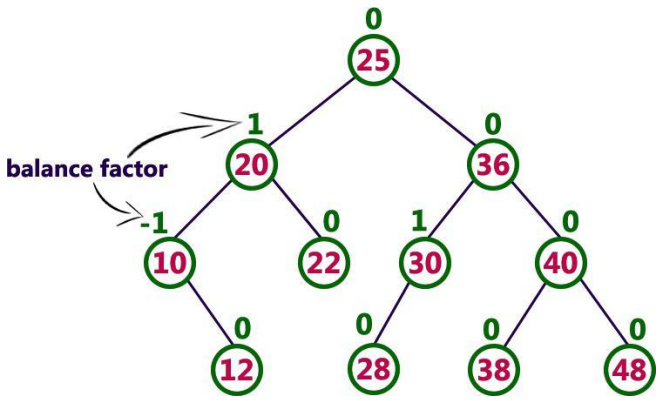
- Tree, Sub-tree, Parent, Child - nodes
- Structure definition for a tree node
- Dynamic memory allocation
- Balancing factor – {-1,0,1}
- Self balancing tree is AVL Tree.

Pre-Lab Task:

AVL tree is a height balanced binary search tree. A binary tree is said to be balanced if, the difference between the heights of left and right subtrees of every node in the tree is either -1, 0 or +1. In an AVL tree, every node maintains an extra information known as balance factor**.**

Balance factor = heightOfLeftSubtree – heightOfRightSubtree

Most of the BST operations (e.g., search, max, min, insert, delete. etc) take O(h) time where h is the height of the BST.The cost of these operations may become O(n) for a skewed Binary tree. If we make sure that height of the tree remains O(Logn) after every insertion and deletion, then we can guarantee an upper bound of O(Logn) for all these operations. Look at a balancing tree- AVL tree.



AVL Tree Rotations:

- In AVL tree, after performing operations like insertion and deletion we need to checkthe **balance factor** of every node in the tree.
- If every node satisfies the balance factor condition, then we conclude the operationotherwise we must make it balanced.
- Whenever the tree becomes imbalanced due to any operation, we use **rotation** operations to make the tree balanced.

1. Construct an AVL Tree with the following data 50, 70, 76, 85, 92, 73, 101, 9, 108, 23, 54, 98, 39, 63, 12, 81.

Solution:-

2. Suppose the numbers 7, 5, 1, 8, 3, 6, 0, 9, 4, 2 are inserted in that order into an initially empty binary search tree. The binary search tree uses the reversal ordering on natural numbers i.e. 9 is assumed to be smallest and 0 is assumed to be largest. The in-order traversal of the resultant binary search tree is

Solution:-

In-Lab Task:

1. AVL Tree

Write a program for AVL tree having functions for the following operations:
• Insert an element (no duplicates are allowed),
• Delete an existing element,
• Traverse the AVL (in-order, pre-order, and post-order

https://www.codechef.com/problems/UCS616A2

Program:-

2. Chef and Tree:

Chef likes trees a lot. Now he thinking about one interesting problem on trees. He finds the following problem interesting, and you?

Given a tree T and Q queries to it. Tree T has N nodes numbered from 1 to N. Each node v has a value Av associated with it. Queries can be one of three types.

• **1 u v** - reverse the values on the path between vertices u and v.

• **2 ul ur vl vr** - print "Yes" if all the values on the path from ul to ur are exactly same as that of path from vl to vr, and "No" otherwise. It is ensured that length of both the paths are same.

• **3 ul ur vl vr** - Copy all values on path from ul to ur and insert them into the path from vl to vr. It is ensured that length of both the paths are same

https://www.codechef.com/problems/CHEFTRE

Program:-

3. Chef and Average on a Tree

Chef has a tree with N nodes. Each node of the tree has an integer weight associated with it.
Let's define the cost of a sequence of numbers as the arithmetic mean of all elements of the sequence.
Next, let's define the cost of a path in the tree as the cost of the sequence of weights of all nodes belonging to the path. (It's possible for a path to contain only one node.)
A set of paths in the tree is called a correct path decomposition if each node of the tree belongs to exactly one of the paths from this set. The cost of a correct decomposition is defined as the minimum of costs of all paths in this decomposition.
Chef would like to find the maximum cost of a correct decomposition. Can you help him?
Link: https://www.codechef.com/problems/L56AVG

Program:-

Post-Lab Task:

1. List out various rotations need for AVL tree construction and their importance. Construct an AVL tree for the values 67, 23, 12, 75, 89, 19, 23, 25, 101.
    i. Height of 19.
    ii. Delete 67.
    iii. In-order successor.

Solution:-

2. Write a c program to implement the following operations on AVL Tree.
    i. search a node.
    ii. Find Maximum.
    iii. Pre-order traversal.

Solution:-

Skill Session:

1. Given n number each representing value of a node, construct a tree in the following manner:
Let's say node to be inserted has value x
If x is less than or equal to value of the current root node then insert to the left of the current root
If x is greater than the value of the current root then insert node in the right of this node
The above insertion will make a special tree known as Binary Search Tree. Determine whether the tree is balanced or not. A tree is said to be balanced if the absolute difference between the depth of left subtree and the right subtree of each node does not differ by more than 1.

Link: https://www.hackerearth.com/problem/algorithm/balanced-tree/

Program:-

2. Construct AVL Tree for the following sequence of numbers - 52 , 64 , 76 , 5 , 18 , 33 , 55 , 34 , 11 ,20, 48

Solution:-

3. Construct an AVL tree with the following node values: 13, 10, 7, 30, 35, 5, 3 and identify the single left rotations

Solution:-

4. Explain AVL Tree and advantages of AVL Tree in the data organization. What is a balancing factor? And its importance to improve the efficiency of tree organization. Representing AVL tree as balanced binary search tree.

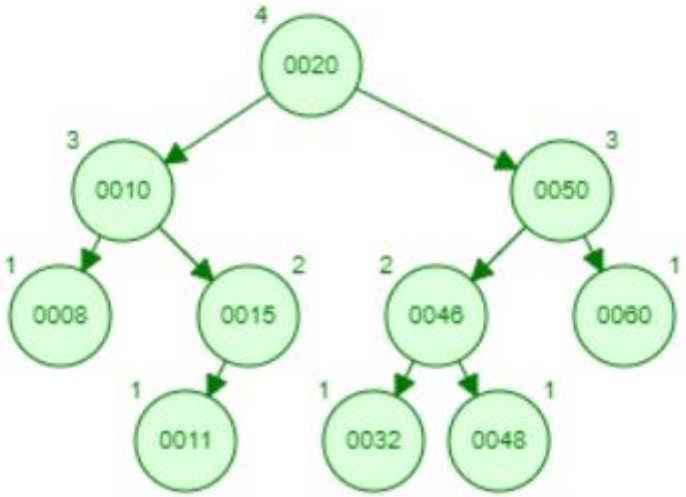
Solution:-

5. Construct AVL Tree for the following sequence of numbers -50 , 20 , 60 , 10 , 8 , 15 , 32 , 46 , 11 , 48. Explain the process of deleting the node 32.

Solution:-

6. Construct AVL Tree for the following sequence of numbers -
50 , 20 , 60 , 10 , 8 , 15 , 32 , 46 , 11 , 48

Output:



Solution:-

Week-12

Lab Session:

Date of the Session: ___/___/___ **Time of the Session: ____to____**

Prerequisite

- Non-linear data structure
- Graphs having edges and nodes
- Structure definition for nodes
- Traversals of a graph
- Shortest path algorithm

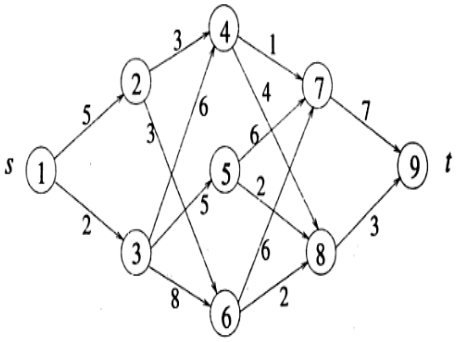
Pre Lab Task

1.Build directed graph that corresponds to this adjacency matrix:

	0	1	2	3
0	true	false	true	false
1	true	false	false	false
2	false	false	false	true
3	true	false	true	false

Solutioin:

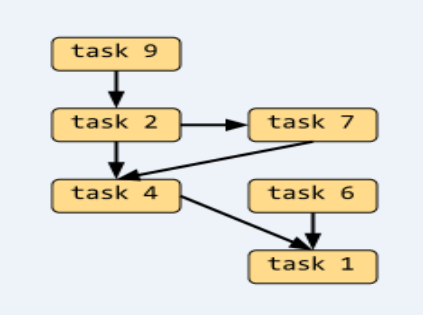
2. Apply breadth-first and depth-first traversal algorithms for the given graph.



Solution:

In Lab Task

1. Analyze why topological sorting can be applicable to Directed acyclic Graphs(DAG).Apply topological sorting for the given graph

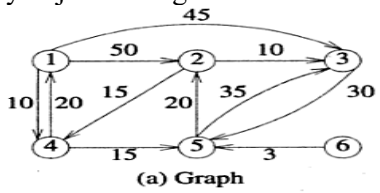


2. Construct the weighted directed graph represented by the adjacency matrix given below. A non-zero value at [row, column] indicates that the vertex in the row is adjacent to the vertex in the column and apply DFS to the graph

	A	B	C	D	E
A	0	5	8	0	0
B	3	0	6	0	0
C	0	3	4	1	0
D	0	6	7	0	0
E	0	0	0	0	0

Solution:-

3. Apply Dijkstra's algorithm to find shortest path from node1 to all other nodes for the following graph.

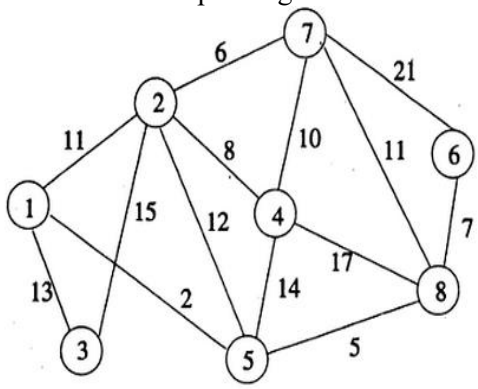


(a) Graph

Solution:-

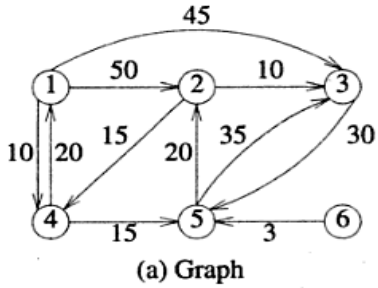
Post-Lab Task

1. Construct a minimal spanning tree for the following graph using Prim's algorithm.



Solution:-

2. Apply krushkal's algorithm and find minimum spanning tree for the following Graph.



(a) Graph

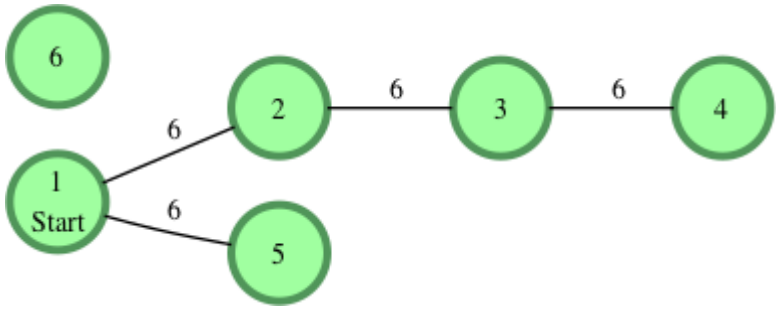
Solution:-

Skill Session:

1) BFS: Shortest Reach in a Graph

Reference Link: https://www.hackerrank.com/challenges/ctci-bfs-shortest-reach/problem

Consider an undirected graph consisting of n nodes where each node is labeled from 1to n and the edge between any two nodes is always of length 6. We define node s to be the starting position for a BFS. Given a graph, determine the distances from the start node to each of its descendants and return the list in node number order, ascending. If a node is disconnected, it's distance should be -1.For example, there are n=6 nodes in the graph with a starting node . The list of edges = [{1,5},{1,2},{2,3},{3,4}] and each has a weight of 6.



Starting from node 1and creating a list of distances, for nodes 2 through 6 we have distance =[6,12,18,6,-1]

. Function Description

Define a Graph class with the required methods to return a list of distances.

Input Format

The first line contains an integer, q, the number of queries. Each of the following q sets of lines is as follows:

The first line contains two space-separated integers, n and m, the number of nodes and the number of edges.

Each of the next lines contains two space-separated integers, u and v, describing an edge connecting node u to node v.

The last line contains a single integer, s , the index of the starting node.

Output Format

For each of the q queries, print a single line of n-1 space-separated integers denoting the shortest distances to each of the n-1 other nodes from starting position s. These distances should be listed sequentially by node number (i.e.,1,2 ,. n), but should not include node . If

some node is unreachable from s , print -1as the distance to that node.

Program:-

2. **DFS Edges:**

Let G be a connected, directed graph with vertices numbered from 1 to N such that any vertex is reachable from vertex 1. In addition, any two distinct vertices U and V are connected by at most one edge (U,V) .

Consider the standard *DFS* (Depth-First Search) algorithm starting from vertex 1 . As every vertex is reachable, each edge (U,V) of is classified by the algorithm into one of four groups:

1. *tree edge*: If V was discovered for the first time when we traversed (U,V) .
2. *back edge*: If V was already on the stack when we tried to traverse .
3. *forward edge*: If V was already discovered *while* U was on the stack.
4. *cross edge*: Any edge that is not a *tree*, *back*, or *forward* edge.

https://www.hackerrank.com/challenges/dfs-edges/problem

Program:-

3. Kruskal (MST): Really Special Subtree:

Given an undirected weighted connected graph, find the Really Special SubTree in it. The Really Special SubTree is defined as a subgraph consisting of all the nodes in the graph and:

There is only one exclusive path from a node to every other node.

The subgraph is of minimum overall weight (sum of all edges) among all such subgraphs.

No cycles are formed

To create the Really Special SubTree, always pick the edge with smallest weight. Determine if including it will create a cycle. If so, ignore the edge. If there are edges of equal weight available:

Choose the edge that minimizes the sum U+V+WT where U and V are vertices and WT is the edge weight.

If there is still a collision, choose any of them.

Print the overall weight of the tree formed using the rules.

https://www.hackerrank.com/challenges/kruskalmstrsub/problem

Program:-

4. Prim's (MST) : Special Subtree:

Given a graph which consists of several edges connecting its nodes, find a subgraph of the given graph with the following properties:

- The subgraph contains all the nodes present in the original graph.
- The subgraph is of minimum overall weight (sum of all edges) among all such subgraphs.
- It is also required that there is **exactly one, exclusive** path between any two nodes of the subgraph.

One specific node is fixed as the starting point of finding the subgraph using Prim's Algorithm.
Find the total weight or the sum of all edges in the subgraph.
https://www.hackerrank.com/challenges/primsmstsub/problem

Program:-

5. **Finding pairs:**

You are given a rooted tree with N nodes and node 1 as a root. There is a unique path between any two nodes. Here, d(i,j) is defined as a number of edges in a unique path between nodes i and j.
You have to find the number of pairs (i, j) such that and $d(i,j)=d(i,1)-d(j,1)$.
https://www.hackerearth.com/practice/algorithms/graphs/depth-first-search/practice-problems/algorithm/find-pairs-4-699bc085/

Program:-

6. **Build a graph :**

You are given an integer n. Determine if there is an unconnected graph with n vertices that contains at least **two** connected components and contains the number of edges that is equal to the number of vertices. Each vertex must follow one of these conditions:
Its degree is less than or equal to **1**.
It's a cut-vertex.

Note

1. The graph must be simple.
2. Loops and multiple edges are not allowed.

https://www.hackerearth.com/practice/algorithms/graphs/graph-representation/practice-problems/algorithm/build-a-graph-5f5c6b4a/

Program:-

(For Evaluator's use only)

Comment of the Evaluator (if Any)	Evaluator"s Observation Marks Secured:_____out of _____ Full Name of the Evaluator: Signature of the Evaluator Date of Evaluation:

THANK YOU