

General Instructions:

Carefully read the following instructions before attempting the paper.

- Except your Name, Roll No, Section and Signature, **DO NOT WRITE** anything on this paper.
- The Exam paper consists of 3 questions on 2 printed sides of 1 page.
- In case of any ambiguity, you may make assumptions, but your assumption must not contradict any statement in the question paper. Also mention your assumptions.
- **DON'T** share your program, if your code is matched to any member of your class, both will get straight F in the course without asking who shared or who magically copied.

Submission Instructions:

- Submit a **.cpp** file for each question.
- All your files must be named by your roll number along with question number e.g. K23-XXXX_Q1.cpp.

Question # 01 (LLO #: 1)**[16 Points, 8 Weightage]**

Ali, a student of BS(AI), has five assignments to complete and submit this week. Each assignment has a specific due day, and Ali wants to organize them in his drawer so that the earliest due assignment is on top. This will allow him to easily find and submit the correct assignment. Help Ali in setting up the drawer with his assignments using a suitable data structure. You need to write a function to sort the assignments in descending order based on their due days (i.e., assignments due later in the week should be on top). Write a function that checks whether a specific assignment is in the drawer and returns its position if present, or returns -1 if not. For example, with 5 assignments: Assignment A (due on 9th October), Assignment B (due on 7th October), Assignment C (due on 5th October), Assignment D (due on 15th October), Assignment E (due on 2nd October). After sorting in descending order, the assignments would be: D, A, B, C, E.

Note: You can use any sorting algorithm except bubble sort and use only binary search for searching.

National University of Computer and Emerging Sciences

Question # 02 (LLO #: 1)

[16 Points, 8 Weightage]

At this point you have studied linked lists very effectively so let's turn this concept into an expression solver. In Mathematics you are given an expression $12+13-5 = 20$ and $12/2*3 = 18$ and you add the resultant it yields 38 so take a linked list with two nodes. In each node there is another linked list (size can vary so assume the expression to be 3 variables) and compute its output. The layout would be something like this mentioned below:

Node1->Node2->NULL

Node1 → 12 → + → 13 → - → 15 → Node1

Node2 → 12 → / → 2 → * → 3 → Node2

Output

38

Note: Both the outer and inner nodes must be a doubly linked lists and order of operations must be PEDMAS (Parenthesis, Exponents, Division, Multiplication, Addition, Subtraction).

Question # 03 (LLO #: 1)

[18 Points, 9 Weightage]

You are tasked of finding strings in an NxN matrix. Using the approach of backtracking you are required to find not just the strings but also their occurrences (if any) in a given matrix.

Example: 5x5matrix

Strings: AD and RADAR

G R O O T
B A R R O
W R D B S
R A D A R
W E S T G

Output:

AD found 2

Radar found 1