Department of Computer Science and Engineering

NATIONAL INSTITUTE OF TECHNOLOGY SRINAGAR

Course: Data Structures
Exam: Mid Term

Max Marks: 30

4th Semester: CSE
Time: 1.5 hrs

Course Outcomes (CO)

CO1	Ability to understand basic data structures and Abstract Data Types along with their definitions and usage in computer science.
CO2	Understanding and implementing linear data structures. Ability to implement both static and dynamic linear data structures like linked lists, stacks and queues. Implementing and understanding the usage of non linear data structures like trees.
CO3	Applications of data structures with some working examples in stacks, queues, linked lists and trees. Memory and storage management concepts.
CO4	Graph representations and algorithms to work on graphs. Sorting, searching algorithms and their implementations using basic data structures.

1. **Answer the following [5 (CO2) ,5 (CO2)]:**
   a. Write a **function** that **takes** the start pointers of two singly linked lists representing two polynomials P and Q and returns the start pointer to the polynomial R which is the difference of the two polynomials P and Q. R is the new polynomial created and **R = P-Q.** Add comments to your code where necessary.
   b. Write a **function** that takes the start pointer to a doubly linked list and an **element**. It searches for this element in the list and **returns** the **position** of the first occurrence of the element as well as **deletes** all the occurrences of the element from the list. Add comments to your code where necessary.

2. **Answer the following [5(CO2) ,5 (CO1)]:**
   a. What are circular queues and why do we implement them? Write down **add and delete functions** for circular queues. Add comments to your code where necessary.
   b. An array ARR [-15.....10, 15.....40] requires **one byte** of storage. If beginning location is 1500 determine the **location of ARR [15][20]** in both **row major and column major** representations.

3. **Answer the following [4 (CO3) ,3(CO3) ,3(CO3)]:**
   a. Convert the following **infix** expression to **postfix** expression.

      { [A * B * C – (D ^ E / F ^ G)] *H } + [K / M *N]

   b. How do stacks help in balancing of symbols? Using expression in part (a) as example, demonstrate in steps how stacks can be used to check if the symbols are balanced.

   c. For the following recursive function FUNC() called from main, show how the system stack grows and shrinks when the value passed by main() to FUNC() is 128. What is the return value to main ()? Show the stack contents and show how the return value is calculated through the recursive function calls.

```
int FUNC(int n)
{
if(n==8)
    return 1;
else
    return ( CALL(n*16) + n/2 );
}
```

```
int CALL ( int n)
{
if( n== 128)
    return 0;
else
    return ( FUNC( n / 32 ) +2 );
}
```