**Program: B. Tech.**     **Branch: CSE+AI+CM+ECM**     **Year: I**     **Semester: II**
**Subject: Data Structures (CS1203)**

**Date: 09/06/2023**                                    **Start Time: 10:00 am**
**Time Duration: 3 Hours**                              **Max. Marks: 100**

## Instructions:

1) All questions are compulsory.
2) Use only pen to write your answers. No marks will be given if an answer is written using a pencil.
3) Ignore syntax errors in code-snippets, if any.

**Q.1 [Marks: 10]** An array A consists of n integers in locations A[0], A[1] ....A[n-1]. It is required to shift the elements of the array cyclically to the left by t places, where $1 <= t <= (n-1)$. An incomplete algorithm for doing this in linear time, without using another array is given below. Complete the algorithm by filling in the blanks. Assume all the variables are suitably declared.

```
min = n; i = 0;

while (_____) {
    temp = arr[i]; j = i;
    while (_____) {
    A[j] = _____
    j= (j + t) mod n;
    if ( j < min ) then
        min = j;
}
arr[(n + i - t) mod n] = _____
i = _____
```

**Q.2 [Marks: 10]** Consider the following function that references a Doubly Linked List head as a parameter. Consider that a node of the doubly linked list has the previous pointer and the next pointer as next.

```
void fun (struct node **head_ref)
{
    struct node *temp = NULL;
    struct node *current = *head_ref;
```

```
while (current! = NULL)
{

    temp = current->prev;
    current->prev = current->next;
    current->next = temp;
    current = current->prev;

}

if (temp != NULL )
    *head_ref = temp->prev;
}
```

Consider that the reference head of the doubly linked list and passed to the above function as (1<->2<->3<->4<->5<->6). Please write modified linked list after the function call?

Q. 3 [Marks: 10] The following function computes the value of $^m C_n$ correctly for all legal values m and n (m$\geq$1, n$\geq$0, and m>n )

```
int func (int m, int n)
{

    if (-------------------)
return 1;
    else
return (func(m -1, n) + func(m - 1, n - 1));
}
```

Please write **condition inside if statement** in terms of m and n to get desired result discussed above.
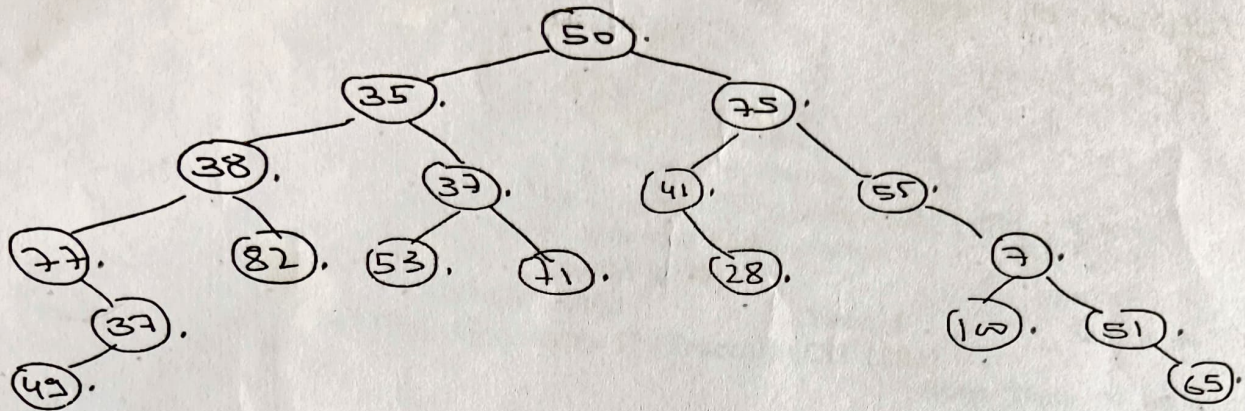
Q.4 [Marks: 7+3] Draw Tree using following traversal

Converse post order: L K G J F C E I H D B A

Normal In-order:     H D I B E A F J C K G L

Find the following:

(1) Draw the tree?
(2) Find converse pre order traversal.

```
                              50
                 35                        75
            38        37              41        55
        77     82   53    71              28        7
          37                                     13   51
        49                                              65
```
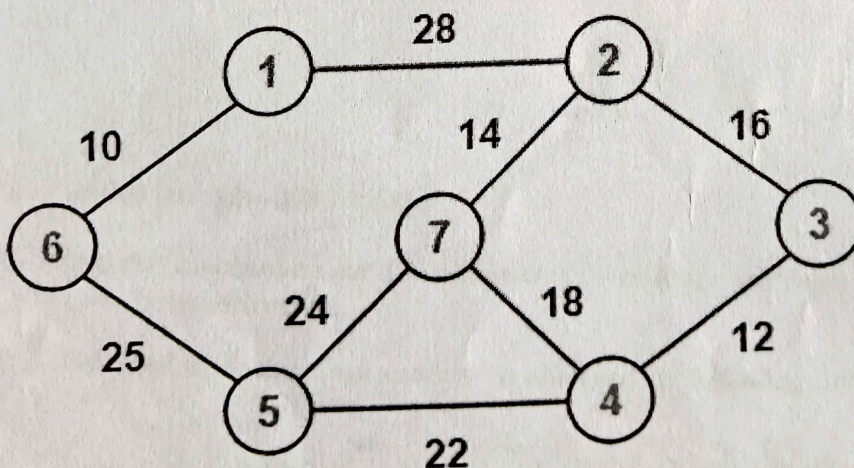
Find following for the above tree.
   (1) In-order Traversal
   (2) Pre-order Traversal
   (3) Post-order Traversal
   (4) Diameter path of tree

Q.6 [Marks: 10] Step-by-step make an AVL tree from the following data.

1, 2, 3, 4, 5, 6, 7, 8, 9 ,10

Q.7 [Marks: 10] We have data 6,14, 9,23, 54, 64, 40 and the hash table size is 10. Using quadratic probing technique insert the above data into table, use module division as hash function. Find number of probe number for each data. Consider 11 as prime number for first hash function and 7 for second hash function.

Q.8 [Marks: 10] Find a minimum spanning tree and total cost using Prim's method step by step for following tree.

```
                      28
          1 ---------------------- 2
        /                          |  \
   10 /            14              | 16 \
    /                              |      \
   6                7              |       3
    \            /     \          |      /
  25 \      24 /         \ 18     | 12 /
      \      /             \      |  /
        5 ---------------------- 4
                  22
```

**Q.9 [Marks: 10]** Find the time complexity of the code snippet.

```
p=0;
For (i=1; i<n; i=i*2)
    {
        P++
    }
    For (j=1; j<p; j=j*2)
    {
        Stmt;

    }
```

**Q.10 [Marks: 7+3]** Write pseudo code or algorithm of Modified Bubble sort also discuss best, worst, and average complexity.