# Data Structure and Algorithms Lab

**Code: PMDS605P**

## Digital Assignment 4

**Name: Soumyadeep Ganguly**

**Reg. No.: 24MDT0082**

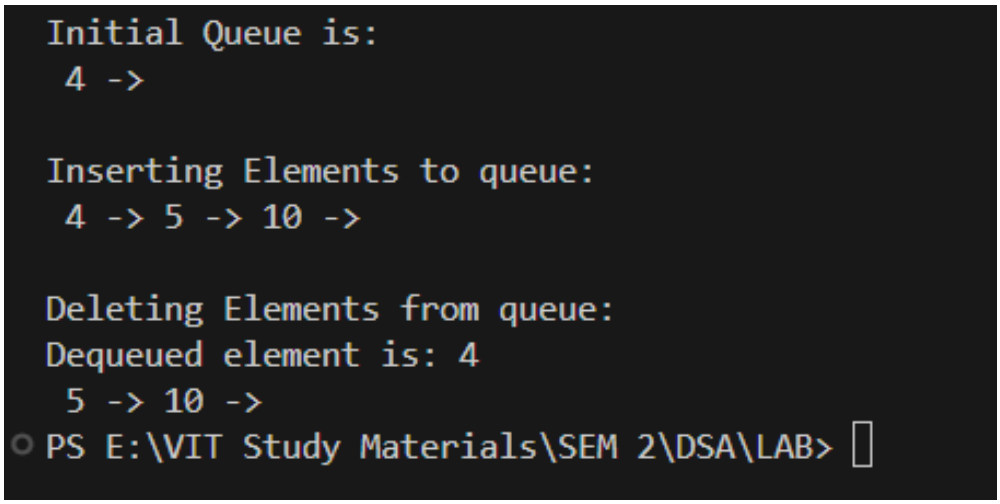
**Course: M.Sc. in Data Science**

## Implement Queue using Linked List in C.

```c

#include<stdio.h>
#include<stdlib.h>

struct queue
{
    int data;
    struct queue * next;
};

struct queue * createNode(){
    return (struct queue*)malloc(sizeof(struct queue));
}

struct queue * enqueue(struct queue * head, int data){
    struct queue * ptr = createNode();
    struct queue * p = head;
    ptr->data = data;
    while(p->next != NULL)
    {
        p = p->next;
    }
    p->next = ptr;
    ptr->next = NULL;
    return head;
}

struct queue * dequeue(struct queue * head){
    struct queue * ptr = head;
    head = head->next;
    int x = ptr->data;
    printf("Dequeued element is: %d \n", x);
    free(ptr);
    return head;
}
```

```c
void traverseQueue(struct queue * ptr){
    if(ptr != NULL){
        printf(" %d ->", ptr->data);
        traverseQueue(ptr->next);
    }
}

int main(){

    struct queue * head;
    head = createNode();
    head->data = 4;
    head->next = NULL;

    printf("Initial Queue is: \n");
    traverseQueue(head);

    printf("\n\n");

    printf("Inserting Elements to queue: \n");
    head = enqueue(head, 5);
    head = enqueue(head, 10);
    traverseQueue(head);

    printf("\n\n");

    printf("Deleting Elements from queue: \n");
    head = dequeue(head);
    traverseQueue(head);

    return 0;

}
```

**OUTPUT:**


Initial Queue is:
4 ->
Inserting Elements to queue:
4 -> 5 -> 10 ->
Deleting Elements from queue:
Dequeued element is: 4
5 -> 10 ->
PS E:\VIT Study Materials\SEM 2\DSA\LAB>