Q. Write a program to implement stacks using linked list:

Program

```
#include <stdio.h>

struct node
{ int data;
  struct node *next;
};
struct node *top = 0;
void push(int val)
{ struct node *newnode;
  newnode = (struct node *)malloc(sizeof(struct node));
  newnode->data = val;
  newnode->next = top;
  top = newnode;
}
void display()
{ struct node *temp;
  temp = top;
  if (top == 0) { printf("stack is empty"); }
  else
  { while (temp != 0)
    { printf("%d\n", temp->data);
      temp = temp->next;
    }
  }
}
void pop()
{ struct node *temp;
  temp = top;
```

```
if(top==0){printf("stack underflow");}
else
{ printf("poped element is %d\n", top->data);
  top=top->next;
  free(temp);
}
}
void main()
{ int choice, num;
  printf("Enter the operation:\n1.push\n2.pop\n
          3.display\n");
  while(1){
          printf("Enter operation\n");
          scanf("%d", &choice);
          if(choice==-1)
          { printf("operation completed.\n");
            break;
          }
  else
  { switch(choice)
    {
    case 1: printf("Enter the number:\n");
            scanf("%d", &num);
            push(num);
            break;
    case 2: pop();
            break;
    case 3: display();
            break;
    default: printf("invalid input");
    }
  }
}
```

Date :
Page No :

Output:

enter the operation:
1. push 2. pop 3. display 4. -1 to stop
enter operation
1
Enter the number: 5
enter operation: 1
Enter the number: 6
enter operation: 1
Enter the number: 7
Enter operation: 3
7
6
5
enter operation 2
poped element is 7.
enter operation 2
poped element is 6.
enter operation 2
poped element is 5
enter operation 2
stack underflow

enter operation -1
operation completed

C:\Users\bmsce\Desktop\22cs300\stackII.exe

```
enter the operation:
1.push
2.pop
3.display
4.-1 to stop
enter operation
1
Enter the number:
5
enter operation
1
Enter the number:
6
enter operation
1
Enter the number:
7
enter operation
2
poped element is 7
enter operation
2
poped element is 6
enter operation
2
poped element is 5
enter operation
2
stack underflow
enter operation
```

b Write a program to implement Queues using linked list.

Program

```
#include <stdio.h>

struct node
{ int data;
  struct node *next;
}
struct node *front=0;
struct node *rear=0;
void enque(int x)
 { struct node *newnode;
   newnode=(struct node *)malloc(sizeof(struct node));
   newnode->data=x;
   newnode->next=0;

   if(front==0 && rear==0)
     { front=rear=newnode; }
   else
     { rear->next=newnode;
       rear=newnode;
     }
}

void display()
{ struct node *temp;
  if(front==0 && rear==0)
    { printf("The Queue is empty\n"); }
  else
    { temp=front;
```

```
        while (temp != 0)
        { printf ("%d", temp->data);
          temp = temp->next;
        }
    }
}
void dequeue()
  { struct node *temp;
    temp = front;
    if (front == 0 && rear == 0)
      { printf ("Queue underflow\n"); }
    else
      { printf ("dequeued element is %d\n",
                front->data);
        front = front->next;
        free (temp);
      }
  }
void main()
{ int choice, num;
  printf("Enter the operation:\n1. enque\n2. deque.\n3. display\n4. -1 to stop\n");
  while(1)
  { printf("Enter operation:\n");
    scanf("%d", &choice);
    if (choice == -1)
    { printf("operation completed\n");
      break;
    }
    else
    { switch (choice)
      {
```

```
        case 1: printf("Enter the numbers:\n");
                scanf("%d", &num);
                enqueue(num);
                break;
        case 2: dequeue();
                break;
        case 3: display();
                break;
        default: printf("invalid input\n");
    }
  }
 }
}
```

29/1/24

//output:

enter the operation

1. enque 2. dequeue 3. display 4. -1 to stop

enter operation 1

enter the number 5

enter operation 1

enter the number 6

enter operation 1

enter the number 7

enter operation 3

5

6

7

enter operation 2

dequeued element is 5

enter operation 2

dequeued elemt is 6

1a. write

linked

Demonstrate

Lecture

29/1/24

C:\Users\bmsce\Desktop\22cs300\queuell.exe

```
enter the operation:
1.enqueue
2.dequeue
3.display
4.-1 to stop
enter operation
1
Enter the number:
5
enter operation
1
Enter the number:
6
enter operation
1
Enter the number:
7
enter operation
3
5
6
7
enter operation
2
deueued element is 5
enter operation
2
deueued element is 6
enter operation
2
deueued element is 7
enter operation
2
queue underflow
```