NAME: Venkateswar L
BRANCH: Computer Science and Engineering SEC: F
ROLL NO.: 230701376

PROGRAM: Implementation Of Queue Using Array and Linked List Implementation

Write a C program to implement a Queue using Array and linked List implementation and execute the following operation on stack.

1. Enqueue
2. Dequeue
3. Display the elements in a Queue

ARRAY IMPLEMENTATION

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

#define size 5
int queue[size];
int f=-1;
int r=-1;

int isfull()
{
   if (r==size-1)
   return 1;
   else
   return 0;
}

void enq(int rol)
{
   if (isfull())
   {
      printf("Overflow!");
   }
   else
   {
      if (f==-1)
         f=0;
      r=r+1;
      queue[r]=rol;
   }
}
```

NAME: Venkateswar L
BRANCH: Computer Science and Engineering SEC: F
ROLL NO.: 230701376

```
int isempty()
{
   if (f==-1)
   return 1;
   else
   return 0;
}

int isreset()
{
   if (f==r)
   return 1;
   else
   return 0;
}

void deq()
{
   if (isempty())
   {
      printf("Underflow!");
   }
   else
   {
      if (f==-1)
         f=0;
      int data;
      data=queue[f];
      f=f+1;
      if (isreset())
         f=-1;
         r=-1;
      printf("\n%d has been dequeued!",data);
   }
}

void display()
{
   int temp=f;
   while(temp<size)
   {
      printf("%d ",queue[temp]);
```

```
        temp++;
    }
}


int main()
{
    while (1)
    {
        printf("Enter a operation to be executed:\n");
        printf("1. Queue\n2. Dequeue\n3. Display\n");
        int r;
        scanf("%d",&r);

        switch(r)
        {
            case 1:
            printf("\nFor Queue: ");
            int y;
            scanf("%d",&y);
            int i=0;
            do{
                printf("\nEnter elements to be queued: ");
                int e;
                scanf("%d",&e);
                enq(e);
                i++;
            }while(i<y);
            display();
            break;

            case 2:
            printf("\nFor Dequeue: ");
            deq();
            break;

            case 3:
            printf("\nCurrent Queue is: ");
            display();
            break;
        }
    printf("\nOperations terminated!\nDo you wish to continue? 1/0\n");
    int ch;
```

```
   scanf("%d",&ch);
   if (ch==1)
   continue;
   else


}
```

LINKED LIST IMPLEMENTATION

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

#define size 5
int queue[size];
int f=-1;
int r=-1;

struct node
{
   int data;
   struct node *link;
}*first=NULL, *rear=NULL;


int isempty()
{
   if (first==NULL && rear==NULL)
   return 1;
   else
   return 0;
}

void enq(int rol)
{
   struct node *newnode, *temp;
   newnode=(struct node *)malloc(sizeof(struct node));
   newnode->data=rol;

   temp=first;
   if (isempty())
   {
      newnode->link=NULL;
```

```
        first=newnode;
        rear=newnode;
    }
    else
    {
        while(temp!=NULL)
        {
            temp=temp->link;
        }
        newnode->link=NULL;
        rear->link=newnode;
        rear=newnode;
    }
}

void deq()
{
    struct node *temp;
    temp=first;
    if(isempty())
    {
        printf("Queue is empty");
    }
    if (first==rear)
    {
        first=rear=NULL;
    }
    else
    {
        printf("\nThe deleted element is: %d",temp->data);
        first=first->link;
        free(temp);
    }
}

void display()
{
    struct node*temp=first;
    if(isempty())
    {
        printf("underflow");
    }
    else
```

```
    {
        while(temp!=NULL)
        {
            printf("%d ",temp->data);
            temp=temp->link;
        }
    }
}

int main()
{
    while(1)
    {
        printf("Enter an operation to be executed: \n");
        printf("1. Enqueue\n2. Dequeue\n3. Display\n");

        int t;
        scanf("%d",&t);
        switch (t)
        {
            case 1:
            printf("\nEnter number of elements to be queued: ");
            int y;
            scanf("%d",&y);
            int i=0;
            while(i<y)
            {
                printf("\nEnter element to be queued: ");
                int e;
                scanf("%d",&e);
                enq(e);
                i++;
            }
            display();
            break;

            case 2:
            printf("\nDequeue operation: ");
            deq();
            break;

            case 3:
            printf("\nCurrent queue is: ");
```

```
            display();
            break;
        }
    printf("\nOperations terminated!\nDo you wish to continue? 1/0\n");
    int ch;
    scanf("%d",&ch);
    if (ch==1)
    continue;
    else
    break;
    }
}
```