Assignment: 7

Name: Ujjwal Kumar Jha

Registration No. 20194196

Group: C_2

1. WAP to implement a stack using queues

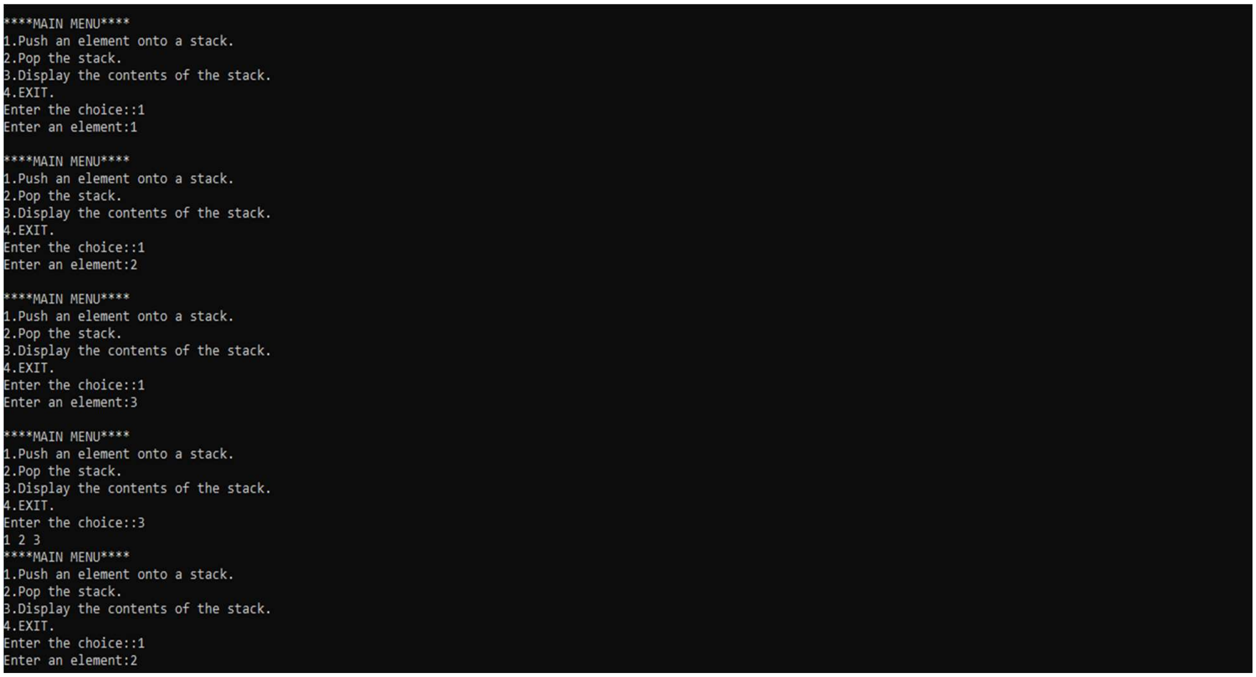
```
#include <stdio.h>
#include <malloc.h>
#define queue struct node
queue
{
int data;
queue *next;
};
queue *front1=NULL;
queue *front2=NULL;
queue* push(int x,queue *front)
{
queue newnode=(queue)malloc(sizeof(queue));
newnode->data=x;
if(front==NULL)
{
```

```
front=newnode;

newnode->next=NULL;

}

else

{

queue *p=front;

while(p->next!=NULL)

{

p=p->next;

}

p->next=newnode;

newnode->next=NULL;

}

return front;

}

void pop()

{

if(front1==NULL)

{

printf("UNDERFLOW");

return;

}

while(front1->next!=NULL)
```

```
{
queue *temp=front1;
front1=front1->next;
front2=push(temp->data,front2);
free(temp);
}
free(front1);
front1=NULL;
while(front2!=NULL)
{
queue *temp=front2;
front2=front2->next;
front1=push(temp->data,front1);
free(temp);
}
}
void display()
{
queue *p=front1;
if(front1==NULL)
printf("Stack is empty");
else
{
```

```
while(p!=NULL)

{

printf("%d ",p->data);

p=p->next;

}

}

}

int main()

{

int opt,x;

do

{

printf("\n***MAIN MENU*");

   printf("\n1.Push an element onto a stack.");

   printf("\n2.Pop the stack.");

   printf("\n3.Display the contents of the stack.");

            printf("\n4.EXIT.");

            printf("\nEnter the choice::");

            scanf("%d",&opt);

            switch(opt)

            {

                    case 1:

                            printf("Enter an element:");
```

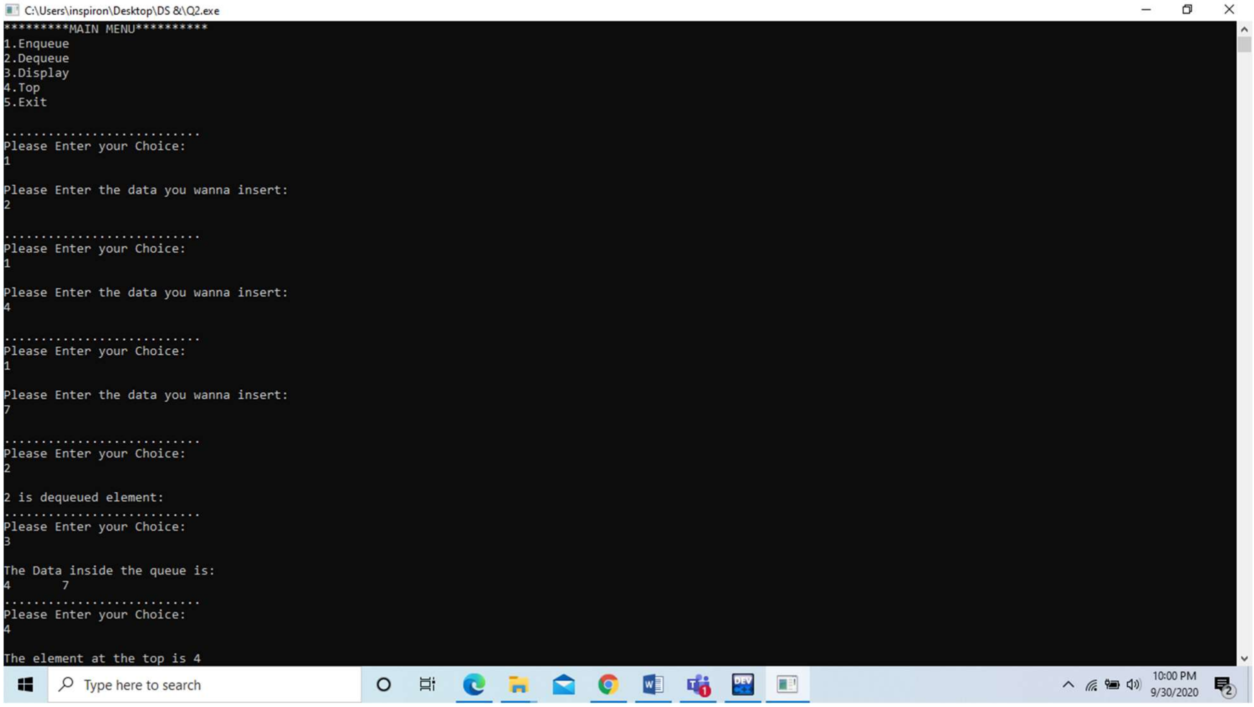
```
                                        scanf("%d",&x);

                                        front1=push(x,front1);

                                        break;

                            case 2:

    pop();

break;

case 3:

    display();

    break;

}

}while(opt!=4);

}
```



```
****MAIN MENU****
1.Push an element onto a stack.
2.Pop the stack.
3.Display the contents of the stack.
4.EXIT.
Enter the choice::1
Enter an element:1

****MAIN MENU****
1.Push an element onto a stack.
2.Pop the stack.
3.Display the contents of the stack.
4.EXIT.
Enter the choice::1
Enter an element:2

****MAIN MENU****
1.Push an element onto a stack.
2.Pop the stack.
3.Display the contents of the stack.
4.EXIT.
Enter the choice::1
Enter an element:3

****MAIN MENU****
1.Push an element onto a stack.
2.Pop the stack.
3.Display the contents of the stack.
4.EXIT.
Enter the choice::3
1 2 3
****MAIN MENU****
1.Push an element onto a stack.
2.Pop the stack.
3.Display the contents of the stack.
4.EXIT.
Enter the choice::1
Enter an element:2
```

Q2: WAP to implement a queues using stack.

```
#include<stdio.h>

#include <stdlib.h>

#define N 5

int s1[N],s2[N],top1=-1,top2=-1,count=0;

void push1(int data){

   if(top1==N-1)

   printf("\nOverflow condition");

   else{

      top1++;

      s1[top1]=data;

   }

}

void push2(int data){

   if(top2==N-1)

   printf("\nOverflow condition");

   else{

      top2++;

      s2[top2]=data;

   }

}

int pop1(){

   if(top1==-1)

   printf("Underflow Condition");

   else

   return s1[top1--];

}

int pop2(){
   if(top2==-1)
   printf("Underflow Condition");
   else
   return s2[top2--];
}
void enqueue(){
   int data;
   printf("\nPlease Enter the data you wanna insert:\n");
   scanf("%d",&data);
   push1(data);
   count++;
}
void top(){
   printf("\nThe element at the top is %d",s1[0]);
}
void deque(){
   int i,a,b;
   if(top1==-1&&top2==-1)
   printf("\nQueue is empty");
   else{
      for(i=0;i<count;i++){
          a=pop1();
          push2(a);
      }
      b=pop2();
      printf("\n%d is dequeued element:",b);
      count--;
      for(i=0;i<count;i++){
          push1(pop2());
      }
   }
}
void display(){
   int i;
   printf("\nThe Data inside the queue is:\n");
   for(i=0;i<=top1;i++){
      printf("%d\t",s1[i]);
   }
}
void main(){
   printf("*********MAIN
MENU**********\n1.Enqueue\n2.Dequeue\n3.Display\n4.Top\n5.Exit\n");
   int n;
   do{
       printf("\n...........................\nPlease Enter your Choice:\n");
       scanf("%d",&n);
       switch (n){
           case 1:
           enqueue();
           break;
           case 2:
           deque();
           break;
           case 3:
           display();
           break;
           case 4:
           top();
           break;
           case 5:
           exit(0);
           break;
           default:
           printf("\nPlease Eter a valid key:");
       }
   }while(n!=5);
}
```



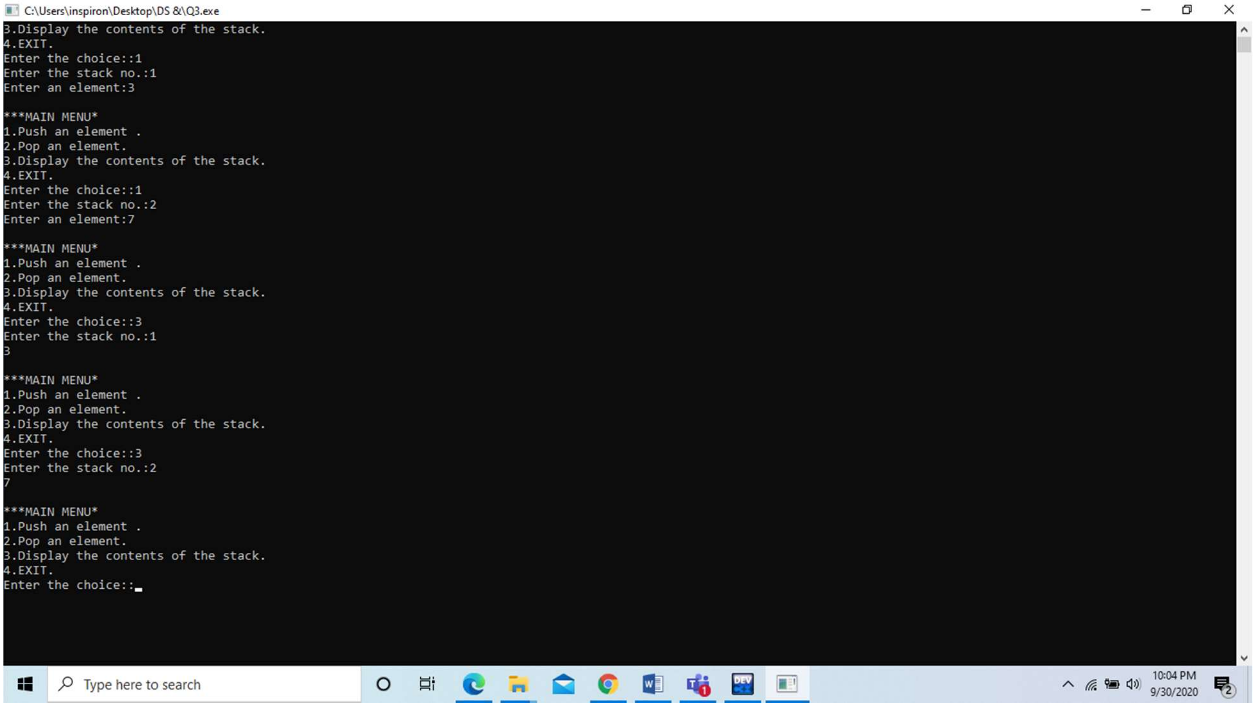
```
*********MAIN MENU**********
1.Enqueue
2.Dequeue
3.Display
4.Top
5.Exit

...........................
Please Enter your Choice:
1

Please Enter the data you wanna insert:
2

...........................
Please Enter your Choice:
1

Please Enter the data you wanna insert:
4

...........................
Please Enter your Choice:
1

Please Enter the data you wanna insert:
7

...........................
Please Enter your Choice:
2

2 is dequeued element:
...........................
Please Enter your Choice:
3

The Data inside the queue is:
4       7
...........................
Please Enter your Choice:
4

The element at the top is 4
```

Q3. WAP to implement n stacks in a single array.

```
#include <stdio.h>
#include <string.h>
#define max 50
int arr[max];
int top[max];
int k;
void push(int sn,int x)
{
if(top[sn-1]==-1)
{
arr[(sn-1)*k]=x;
top[sn-1]=(sn-1)*k;
}
else if(top[sn-1]+1==(sn*k) || top[sn-1]==max-1)
{
printf("Overflow");
}
else
{
top[sn-1]+=1;
arr[top[sn-1]]=x;
}
}
void pop(int sn)
{
```

```
if(top[sn-1]==-1)
{
printf("Underflow");
}
else if(top[sn-1]==(sn-1)*k)
{
top[sn-1]=-1;
}
else
{
top[sn-1]-=1;
}
}
void display(int sn)
{
int i;
if(top[sn-1]==-1)
{
printf("Stack is empty");
return;
}
for(i=top[sn-1];i>=(sn-1)*k;i--)
{
printf("%d\n",arr[i]);
}
}
int main()
{
```

```
int opt,x,n,sn;
memset(top,-1,sizeof(top));
printf("Enter the total no of stacks:");
scanf("%d",&n);
k=max/n;
do
{
printf("\n***MAIN MENU*");
   printf("\n1.Push an element .");
   printf("\n2.Pop an element.");
   printf("\n3.Display the contents of the stack.");
         printf("\n4.EXIT.");
         printf("\nEnter the choice::");
         scanf("%d",&opt);
         switch(opt)
         {
                  case 1:
                           printf("Enter the stack no.:");
                           scanf("%d",&sn);
                           printf("Enter an element:");
                           scanf("%d",&x);
                           push(sn,x);
                           break;
                  case 2:
                           printf("Enter the stack no.:");
                           scanf("%d",&sn);
         pop(sn);
break;
case 3:
            printf("Enter the stack no.:");
                                    scanf("%d",&sn);
    display(sn);
    break;
}
}while(opt!=4);
}
```



```
3.Display the contents of the stack.
4.EXIT.
Enter the choice::1
Enter the stack no.:1
Enter an element:3

***MAIN MENU*
1.Push an element .
2.Pop an element.
3.Display the contents of the stack.
4.EXIT.
Enter the choice::1
Enter the stack no.:2
Enter an element:7

***MAIN MENU*
1.Push an element .
2.Pop an element.
3.Display the contents of the stack.
4.EXIT.
Enter the choice::3
Enter the stack no.:1
3

***MAIN MENU*
1.Push an element .
2.Pop an element.
3.Display the contents of the stack.
4.EXIT.
Enter the choice::3
Enter the stack no.:2
7

***MAIN MENU*
1.Push an element .
2.Pop an element.
3.Display the contents of the stack.
4.EXIT.
Enter the choice::_
```

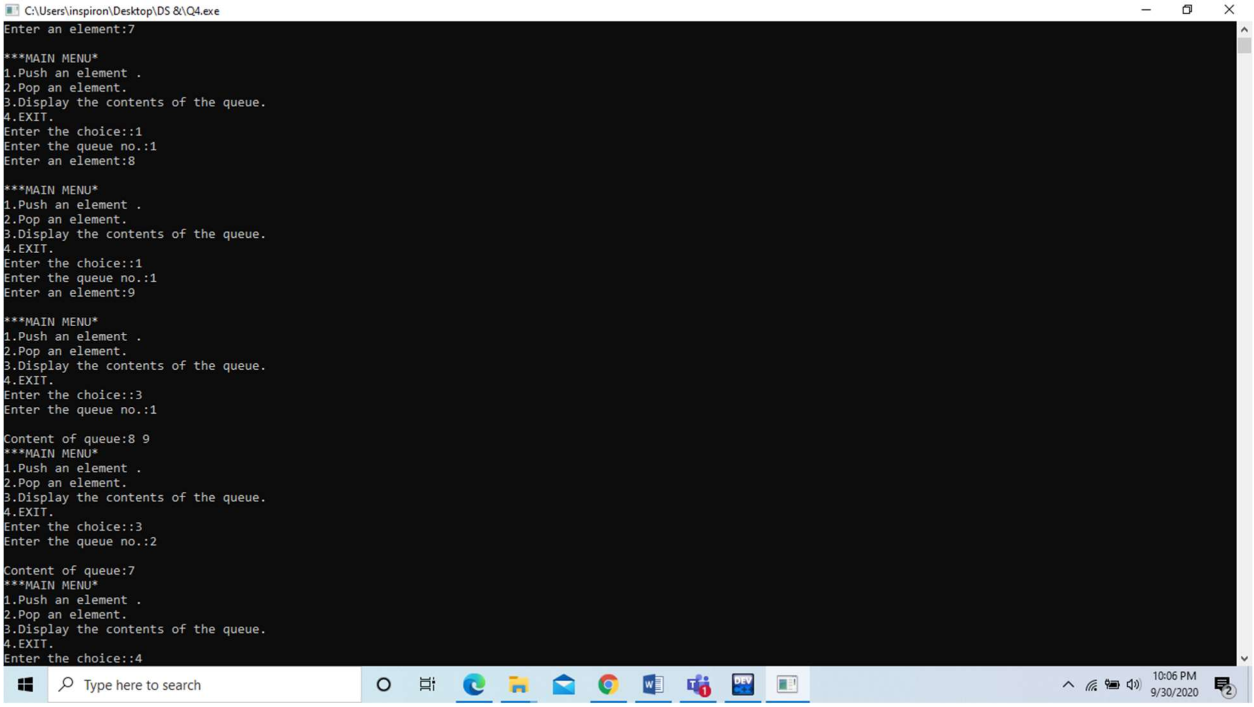
4. WAP to implement n queues in a single array.

```
#include <stdio.h>
#include <string.h>
#define max 50
int arr[max];
int front[max];
int rear[max];
int k;
void push(int qn,int x)
{
 if(front[qn-1]==-1)
 {
        arr[(qn-1)*k]=x;
        front[qn-1]=(qn-1)*k;
        rear[qn-1]=(qn-1)*k;
 }
 else if(rear[qn-1]+1==(qn*k) || rear[qn-1]==max-1)
 {
        printf("OVERFLOW");
 }
 else
 {
        rear[qn-1]+=1;
        arr[rear[qn-1]]=x;
 }
}
void pop(int qn)
{
```

```
if(front[qn-1]==-1)
{
printf("UNDERFLOW");
}
else if(rear[qn-1]==front[qn-1])
{
rear[qn-1]=front[qn-1]=-1;
}
else
{
int i;
for(i=front[qn-1]+1;i<=rear[qn-1];i++)
{
arr[i-1]=arr[i];
}
rear[qn-1]-=1;
}
}
void display(int qn)
{
int i;
if(front[qn-1]==-1)
{
printf("Queue is empty");
return;
}
printf("\nContent of queue:");
for(i=front[qn-1];i<=rear[qn-1];i++)
```

```
{
printf("%d ",arr[i]);
}
}
int main()
{
int opt,x,n,qn;
memset(front,-1,sizeof(front));
memset(rear,-1,sizeof(rear));
printf("Enter the total no of queues:");
scanf("%d",&n);
k=max/n;
do
{
printf("\n***MAIN MENU*");
   printf("\n1.Push an element .");
   printf("\n2.Pop an element.");
   printf("\n3.Display the contents of the queue.");
         printf("\n4.EXIT.");
         printf("\nEnter the choice::");
         scanf("%d",&opt);
         switch(opt)
         {
                  case 1:
                           printf("Enter the queue no.:");
                           scanf("%d",&qn);
                           printf("Enter an element:");
                           scanf("%d",&x);
```

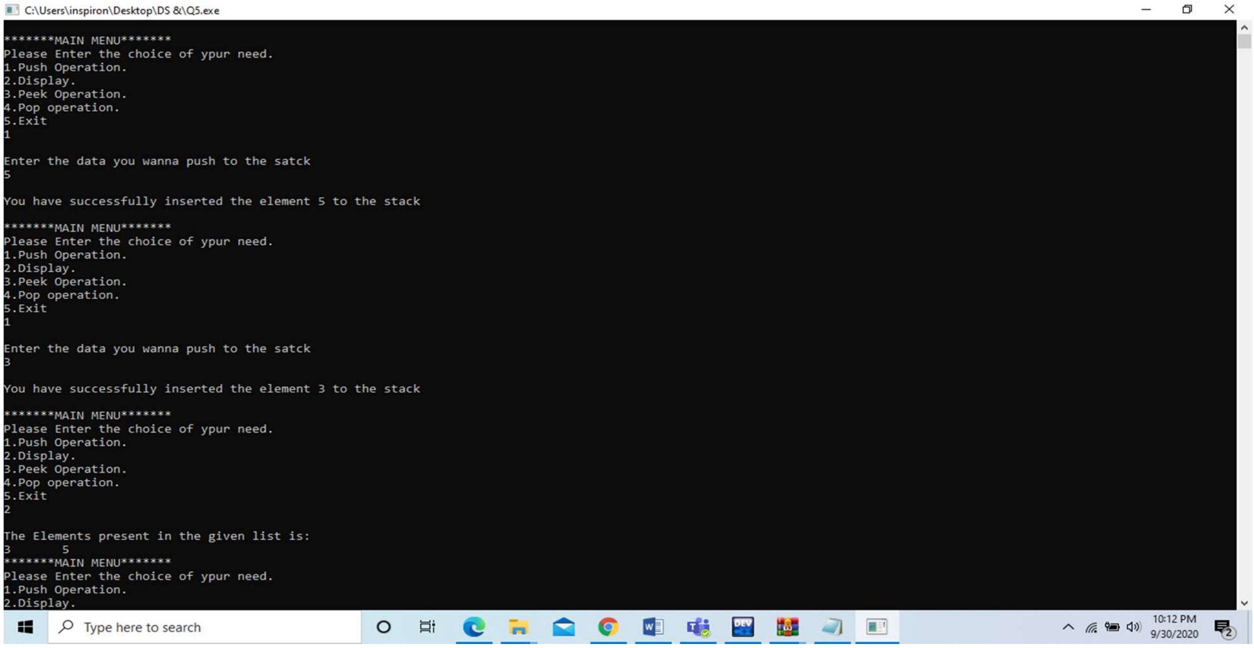
```
            push(qn,x);
            break;
        case 2:
            printf("Enter the queue no.:");
            scanf("%d",&qn);
    pop(qn);
break;
case 3:
    printf("Enter the queue no.:");
            scanf("%d",&qn);
  display(qn);
  break;
}
}while(opt!=4);
}
```



```
Enter an element:7

***MAIN MENU*
1.Push an element .
2.Pop an element.
3.Display the contents of the queue.
4.EXIT.
Enter the choice::1
Enter the queue no.:1
Enter an element:8

***MAIN MENU*
1.Push an element .
2.Pop an element.
3.Display the contents of the queue.
4.EXIT.
Enter the choice::1
Enter the queue no.:1
Enter an element:9

***MAIN MENU*
1.Push an element .
2.Pop an element.
3.Display the contents of the queue.
4.EXIT.
Enter the choice::3
Enter the queue no.:1

Content of queue:8 9
***MAIN MENU*
1.Push an element .
2.Pop an element.
3.Display the contents of the queue.
4.EXIT.
Enter the choice::3
Enter the queue no.:2

Content of queue:7
***MAIN MENU*
1.Push an element .
2.Pop an element.
3.Display the contents of the queue.
4.EXIT.
Enter the choice::4
```

5. WAP to implement Stack using Linked List.

```
#include<stdio.h>
#include<stdlib.h>
struct node{
    int data;
    struct node *next;
}*top=NULL;
void push(int x){
    struct node *ptr;
    ptr=(struct node *)malloc(sizeof(struct node));
    ptr->data=x;
    ptr->next=top;
    top=ptr;
    printf("\nYou have successfully inserted the element %d to the stack\n",x);
}
void display(){
    struct node *temp;
    temp=top;
    if(top==NULL)
    printf("\nList is Empty\n");
    printf("\nThe Elements present in the given list is:\n");
    while(temp!=NULL){
        printf("%d\t",temp->data);
temp=temp->next;
    }
}
void pop(){
    struct node *temp;
    temp=top;
    if(top==0)
    printf("\nUnderflow Condition\n");
    else{
        printf("The Popped eleemnt is %d",top->data);
        top=top->next;
        free(temp);
    }
}
void peek(){
    if(top==NULL){
        printf("\nNo Element Find In Stack\n");
    }
    else{
        printf("\nThe Element at the top is %d",top->data);
    }
}
void main(){
    int n,x;

    do{
        printf("\n*******MAIN MENU*******\nPlease Enter the choice of ypur need.\n1.Push Operation.\n2.Display.\n3.Peek Operation.\n4.Pop operation.\n5.Exit\n");
    scanf("%d",&n);
        switch(n){
            case 1:
                printf("\nEnter the data you wanna push to the satck\n");
                scanf("%d",&x);
                push(x);
                break;
            case 2:
            display();
            break;
            case 3:
            peek();
            break;


            case 4:
            pop();
            case 5:
            exit;
            break;
            default:
            printf("\nWrong input provided\n");


        }
    }while(n!=5); }
```



C:\Users\inspiron\Desktop\DS &\Q5.exe

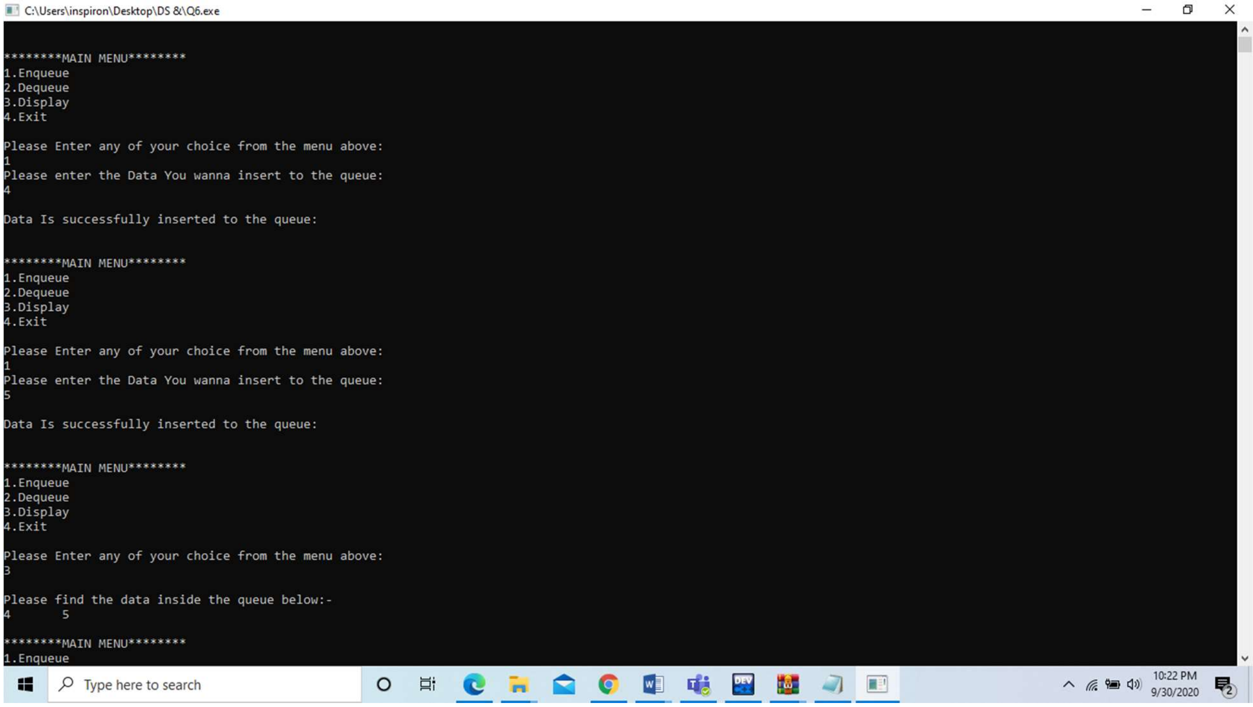
```
*******MAIN MENU*******
Please Enter the choice of ypur need.
1.Push Operation.
2.Display.
3.Peek Operation.
4.Pop operation.
5.Exit
1

Enter the data you wanna push to the satck
5

You have successfully inserted the element 5 to the stack

*******MAIN MENU*******
Please Enter the choice of ypur need.
1.Push Operation.
2.Display.
3.Peek Operation.
4.Pop operation.
5.Exit
1

Enter the data you wanna push to the satck
3

You have successfully inserted the element 3 to the stack

*******MAIN MENU*******
Please Enter the choice of ypur need.
1.Push Operation.
2.Display.
3.Peek Operation.
4.Pop operation.
5.Exit
2

The Elements present in the given list is:
3       5
*******MAIN MENU*******
Please Enter the choice of ypur need.
1.Push Operation.
2.Display.
```

6. WAP to implement Queue using Linked List.

```
#include<stdio.h>
#include<stdlib.h>
struct node{
int data;
struct node *next;
}*front=NULL,*rear=NULL;
void enqueue(){
    int x;
struct node *newnode;
newnode=(struct node *)malloc(sizeof(struct node));
printf("Please enter the Data You wanna insert to the queue:\n");
scanf("%d",&x);
newnode->data=x;
newnode->next=NULL;
if(front==NULL&&rear==NULL)
    front=rear=newnode;
else{
    rear->next=newnode;
    rear=newnode;
}
printf("\nData Is successfully inserted to the queue:\n");
}
void display(){
struct node *temp;
temp=front;
```

```
printf("\nPlease find the data inside the queue below:-\n");
if(front==NULL)
    printf("\nNo Data found\n");
else{
    while(temp!=NULL){
        printf("%d\t",temp->data);
        temp=temp->next;
    }
}
}
void deque(){
struct node *temp;
temp=front;
if(temp==NULL)
    printf("\nUnderflow Condition\n");
else{
    front=front->next;
    free(temp);
}
printf("\nDequeue operation is successfully performed\n");
}
void main(){
    int n;
    do{
printf("\n\n********MAIN
MENU********\n1.Enqueue\n2.Dequeue\n3.Display\n4.Exit\n\nPlease Enter any of your
choice from the menu above:\n");


scanf("%d",&n);
```

```
    switch(n){
    case 1:
        enqueue();break;
    case 2:
        deque();break;
    case 3:
        display();break;


    default:
        printf("Please provide a valid key\n");
    }
}while(n!=4);
}
```



```
********MAIN MENU********
1.Enqueue
2.Dequeue
3.Display
4.Exit

Please Enter any of your choice from the menu above:
1
Please enter the Data You wanna insert to the queue:
4

Data Is successfully inserted to the queue:


********MAIN MENU********
1.Enqueue
2.Dequeue
3.Display
4.Exit

Please Enter any of your choice from the menu above:
1
Please enter the Data You wanna insert to the queue:
5

Data Is successfully inserted to the queue:


********MAIN MENU********
1.Enqueue
2.Dequeue
3.Display
4.Exit

Please Enter any of your choice from the menu above:
3

Please find the data inside the queue below:-
4       5

********MAIN MENU********
1.Enqueue
```