DSA LAB

Week 5

Udeet Mittal

CSE C3

Roll Number 64

1.**SOLVED EXERCISE:**
Implement a queue of integers. Include functions insertq, deleteq and displayq.

Filename: queue_fun.h

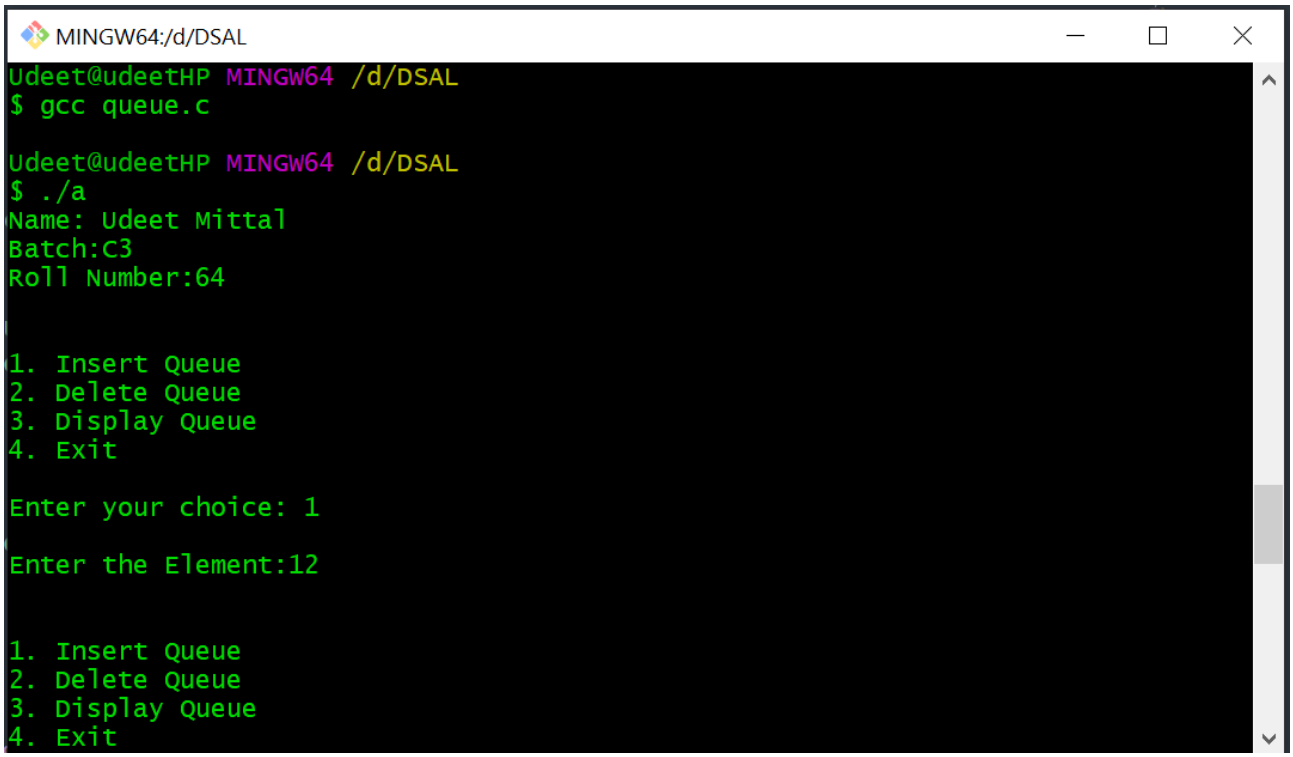
```
#define MAX 20
typedef struct {
   int x[MAX];
   int front;
   int rear;
} queue;

void insertq(queue *, int);
void displayq(queue);
int deleteq(queue *);

void insertq(queue * q,int x)
 {
   if(q->rear==MAX)
   {
      printf("\nOverflow\n");
   }
   else
   {
      q->x[++q->rear]=x;
      if(q->front==-1)
      {
         q->front=0;
      }
   }
 }

 int deleteq(queue * q)
 {
```

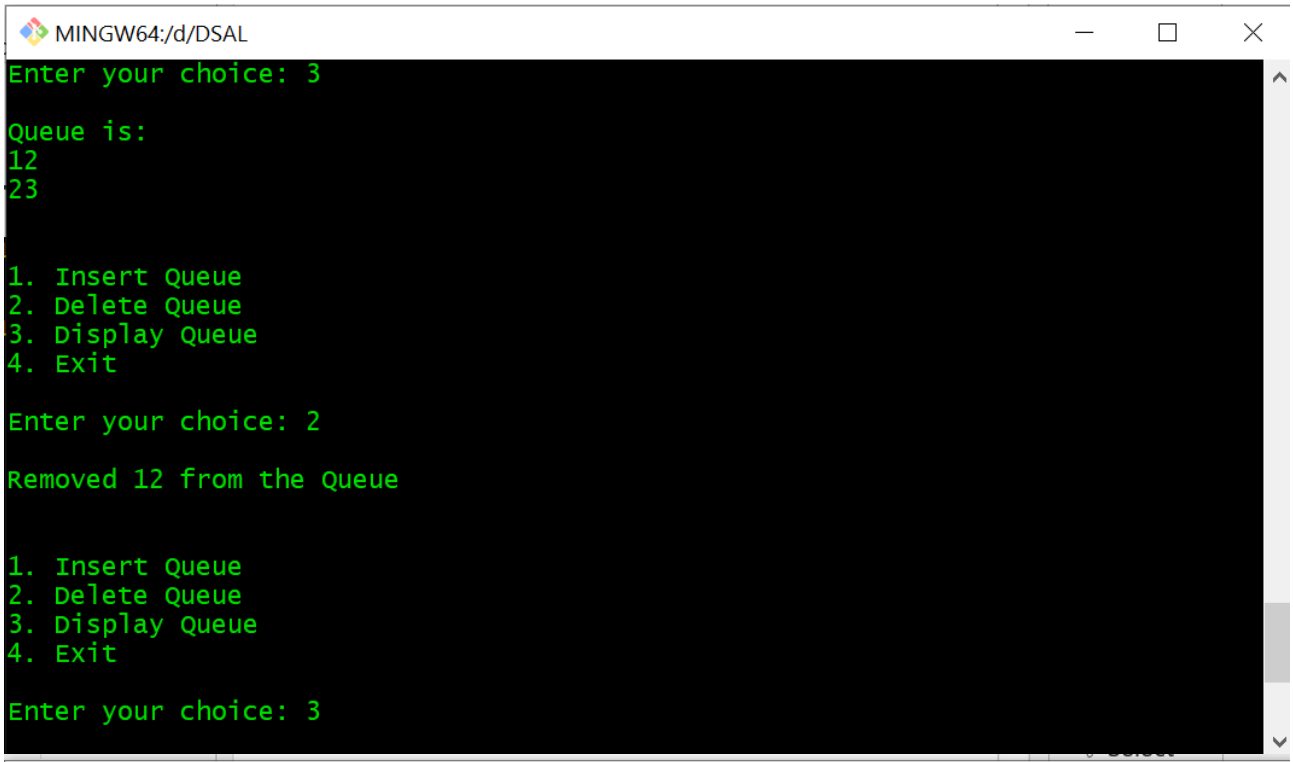
```
    int x;
    if(q->front==-1)
    {
       printf("\nUnderflow!!!\n");
    }
    else if(q->front==q->rear)
    {
       x=q->x[q->front];
       q->front=q->rear=-1;
       return x;
    }
    else
    {
       return q->x[q->front++];
    }
 }

  void displayq(queue q)
 {
    int i;
    if(q.front==-1&&q.rear==-1)
    {
      printf("\nQueue is Empty!!!");
    }
    else
    {
      printf("\nQueue is:\n");
      for(i=q.front;i<=q.rear;i++)
      {
        printf("%d\n",q.x[i]);
      }
    }
}
```

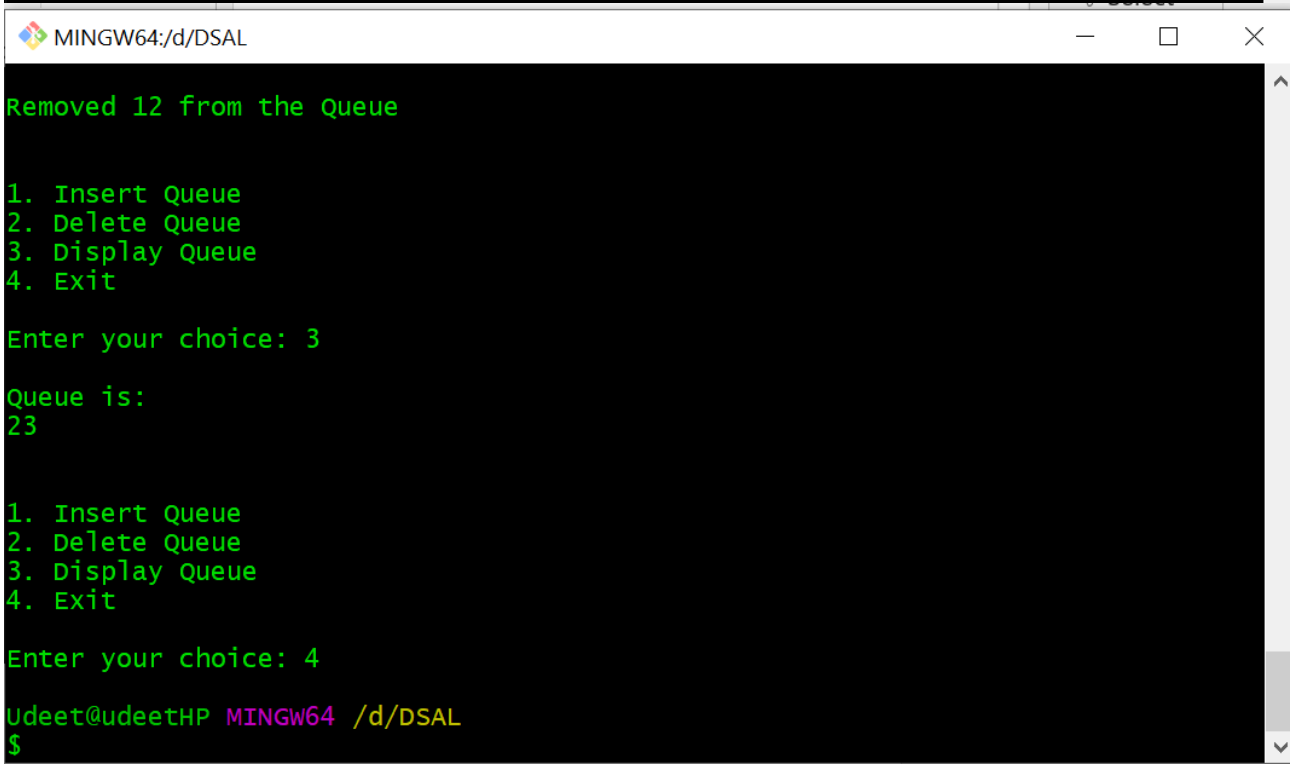
filename: queue.c

```
#include <stdio.h>
#include "queue_fun.h"

int main()
{
   queue q;
   q.front=-1;
   q.rear=-1;
   int ch,x,flag=1;
```

```
printf("Name: Udeet Mittal\nBatch:C3\nRoll Number:64\n");
  while(flag)
    {
      printf("\n\n1. Insert Queue\n2. Delete Queue\n3. Display Queue\n4. Exit\n\n");
       printf("Enter your choice: ");
      scanf("%d",&ch);
      switch(ch)
      {
         case 1:
            printf("\nEnter the Element:");
            scanf("%d",&x);
            insertq(&q,x);
            break;
        case 2:
            x=deleteq(&q);
            printf("\nRemoved %d from the Queue\n",x);
            break;
        case 3:
            displayq(q);
            break;
        case 4:
            flag=0;
            break;
        default:
            printf("\nWrong choice!!! Try Again.\n");
      }
    }
  return 0;}
```







Questions for Lab 5

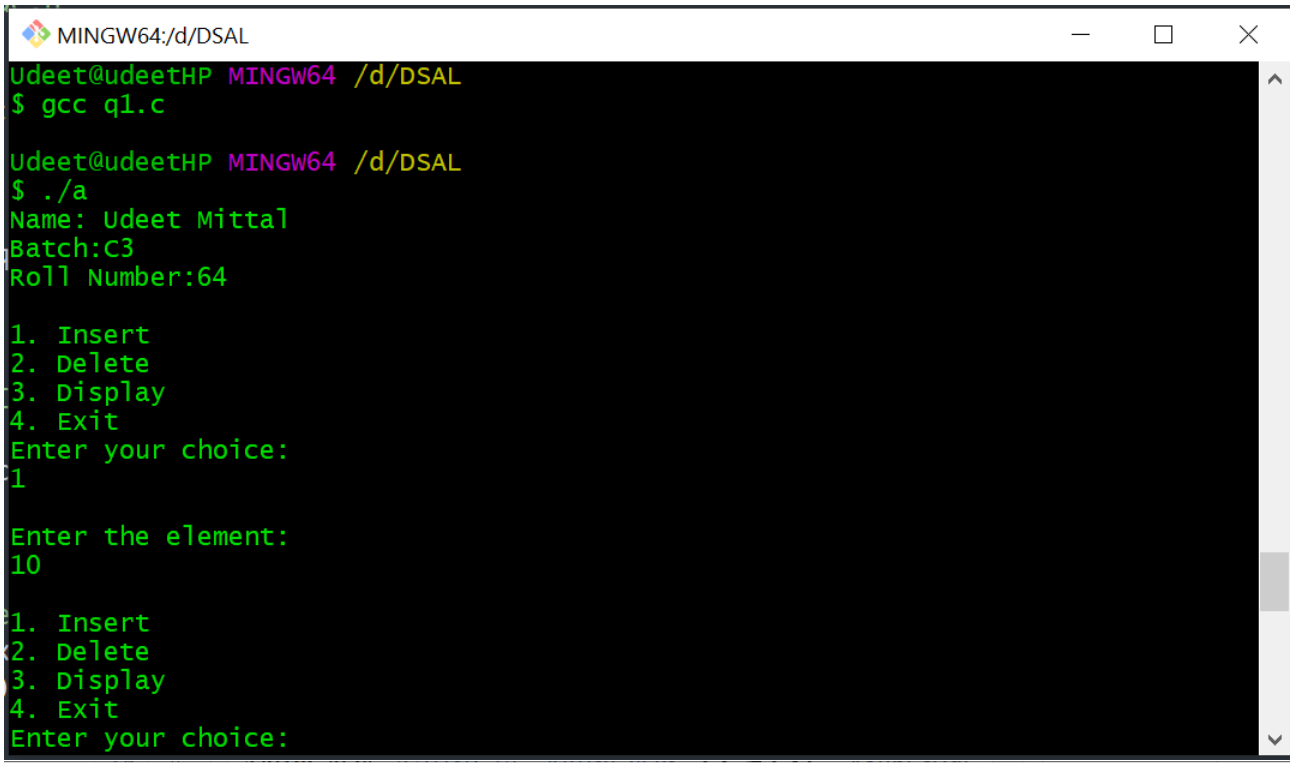
1) Implement a circular queue of integers. Include functions insertcq, deletecq and displaycq.

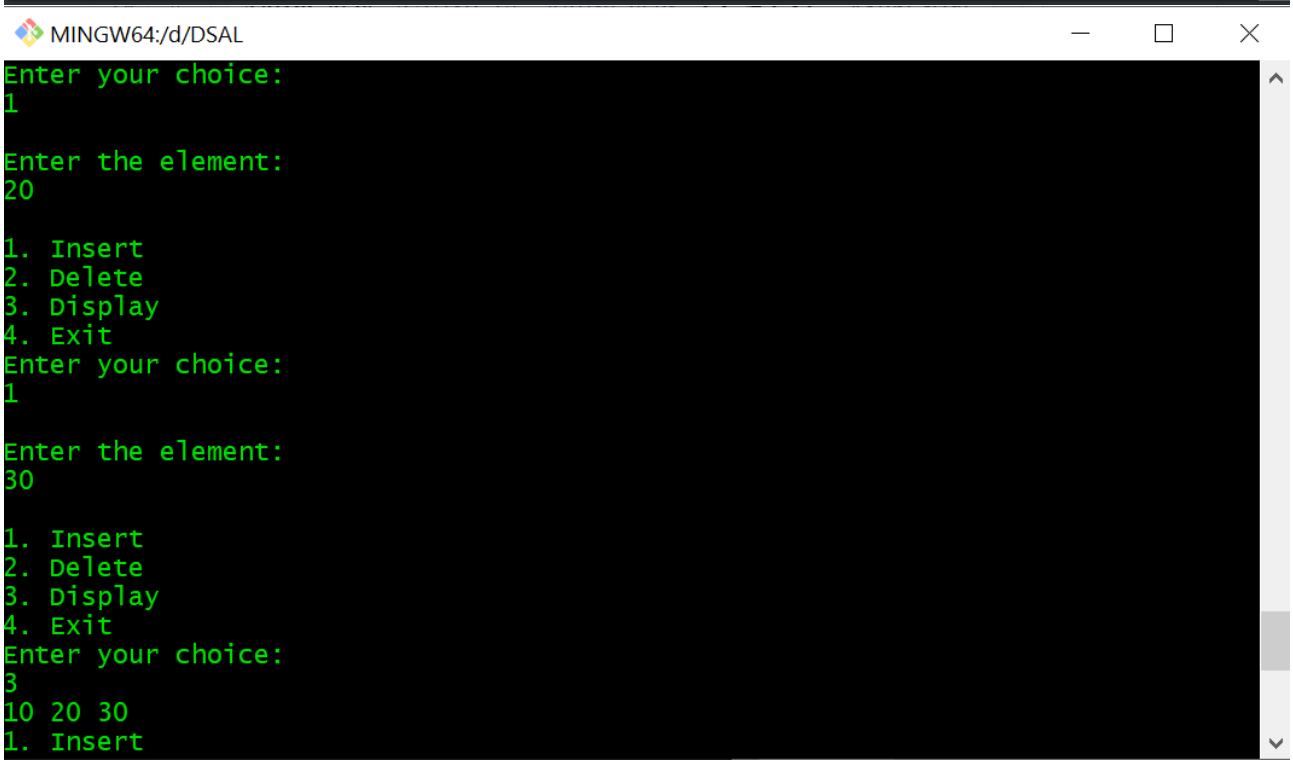
filename: cqueue.h

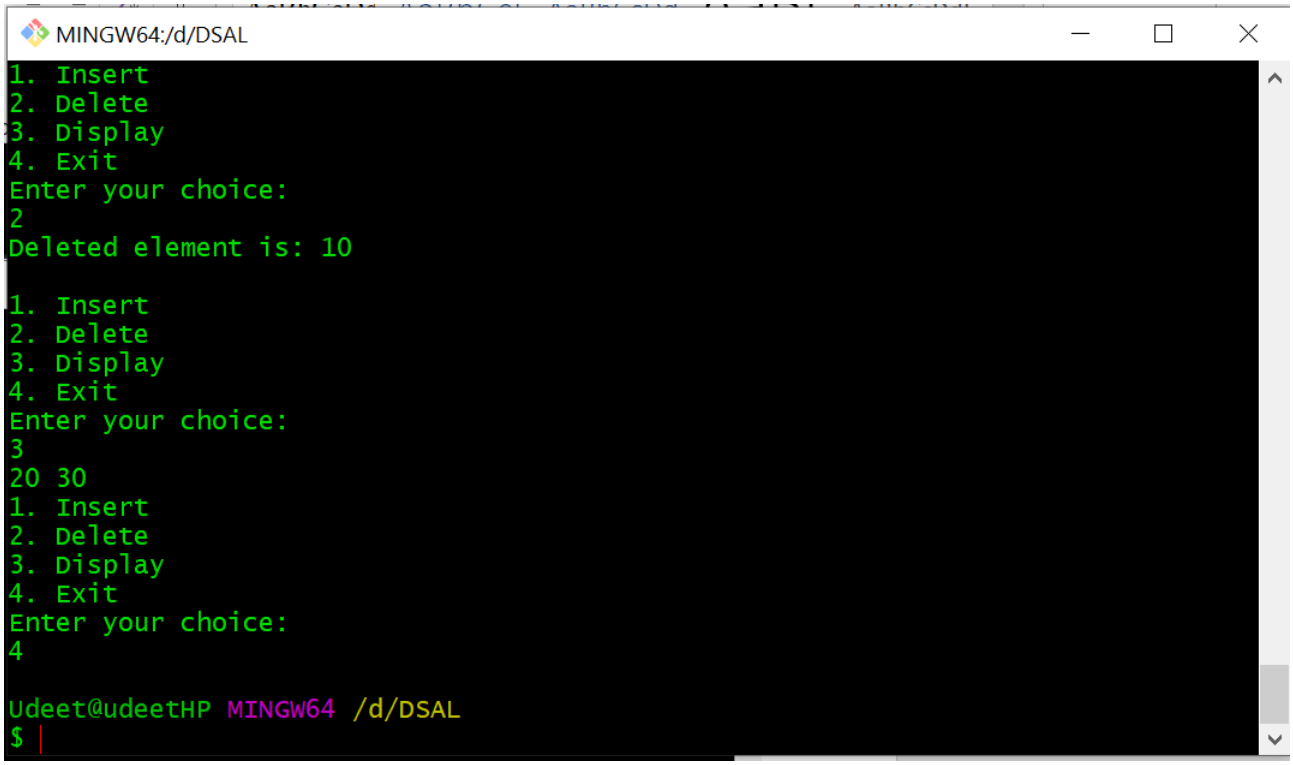
```
#include <stdio.h>
#define max 4
typedef struct
{
 int que[max];
 int front;
 int rear;
}cqueue;
int isfull(cqueue *q)
{
 if ((q->rear + 1) % max == q->front)
 return 1;
 return 0;
}
int isempty(cqueue *q)
{
 if (q->front == q->rear)
 return 1;
 return 0;
}
void addcq(cqueue *q, int ele)
{
 q->rear = (q->rear + 1) % max;
 q->que[q->rear] = ele;
}
void displaycq(cqueue *q)
{
 int i;
 for (i = q->front + 1; i != q->rear; i = (i + 1) % max)
 printf("%d ", q->que[i]);
 printf("%d ", q->que[i]);
}
int deletecq(cqueue *q)
{
 int ele;
 q->front = (q->front + 1) % max;
 ele = q->que[q->front];
 return ele;
}
```

filename: q1.c

```
#include <stdio.h>
#include "cqueue.h"
int main()
{
  printf("Name: Udeet Mittal\nBatch:C3\nRoll Number:64\n");
 cqueue qu;
 qu.front = 0;
 qu.rear = 0;
 cqueue *q = &qu;
 int ch, i, x;
 do
 {
 printf("\n1. Insert\n2. Delete\n3. Display\n4. Exit\n");
 printf("Enter your choice:\n");
 scanf("%d", &ch);
 switch (ch)
 {
 case 1:
 printf("\nEnter the element:\n");
 scanf("%d", &x);
 if (isfull(q))
 printf("Queue is full\n");
 else
 addcq(q, x);
 break;
 case 2:
 if (isempty(q))
 printf("Queue is empty\n");
 else
 {
 x = deletecq(q);
 printf("Deleted element is: %d\n", x);
 }
 break;
 case 3:
 if (isempty(q))
 printf("Queue is empty\n");
 else
 displaycq(q);
 break;
 }
 } while (ch != 4);
 return 0;
}
```


MINGW64:/d/DSAL
Udeet@udeetHP MINGW64 /d/DSAL
$ gcc q1.c
Udeet@udeetHP MINGW64 /d/DSAL
$ ./a
Name: Udeet Mittal
Batch:C3
Roll Number:64
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice:
1
Enter the element:
10
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice:


MINGW64:/d/DSAL
Enter your choice:
1
Enter the element:
20
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice:
1
Enter the element:
30
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice:
3
10 20 30
1. Insert



3.) Implement a queue with two stacks without transferring the elements of the second stack back to stack one. (use stack1 as an input stack and stack2 as an output stack).

filename: q2.h

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 4
typedef struct Stack
{
 int arr[MAX];
 int top;
} Stack;
int isEmpty(Stack *s)
{
```

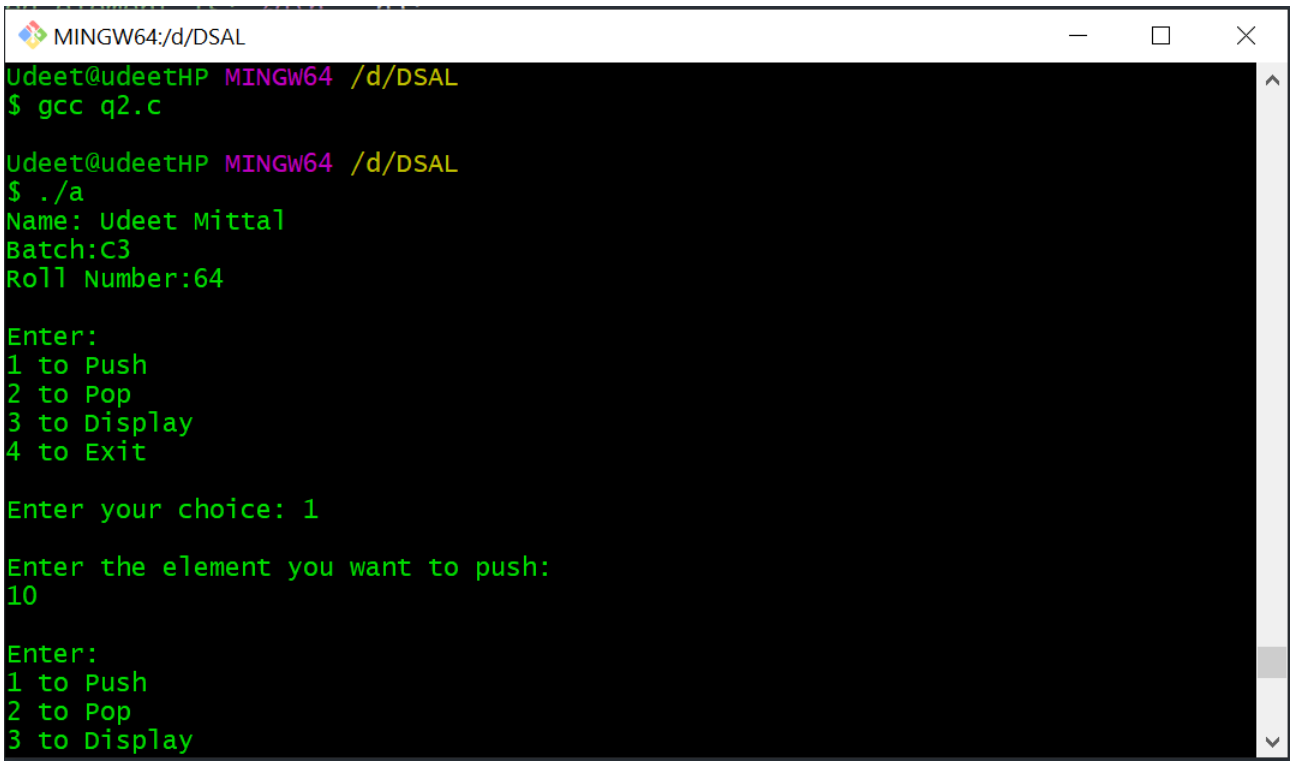
```
 if (s->top == -1)
 return 1;
 return 0;
}
void push(Stack *s, int ch)
{
 if ((s->top + 1) < MAX)
 s->arr[++(s->top)] = ch;
 else
 printf("Overflow!\n");
}
int pop(Stack *s)
{
 if (isEmpty(s))
 return -1;
 return s->arr[(s->top)--];
}
```

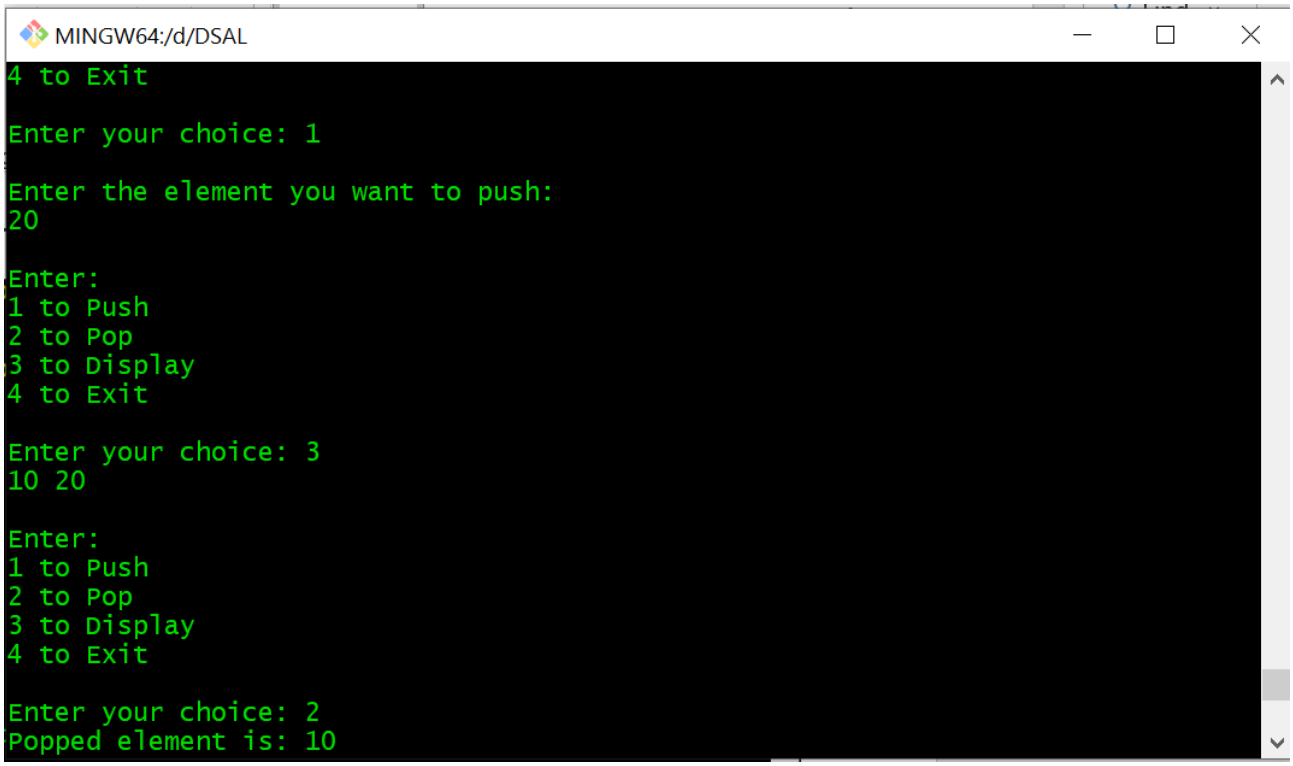
Filename: q2.c

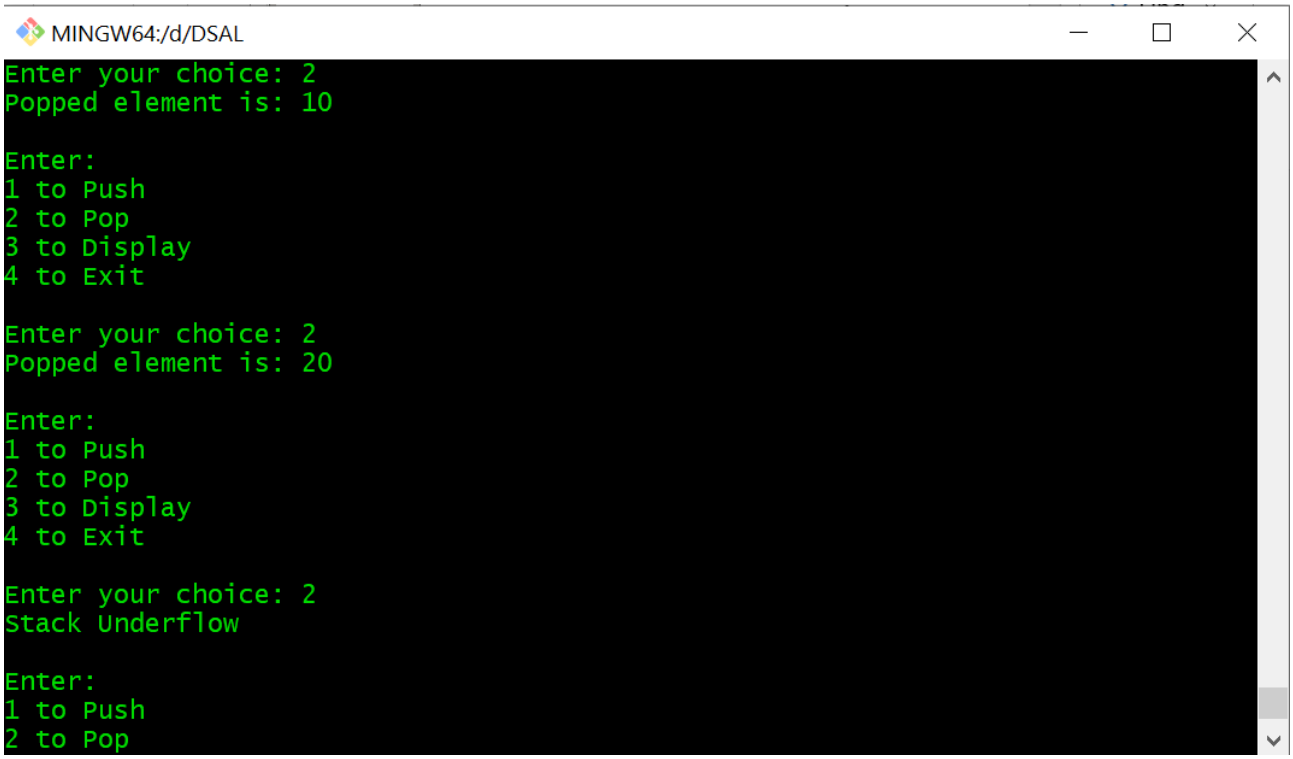
```
#include <stdio.h>
#include "q2.h"
int main()
{
 printf("Name: Udeet Mittal\nBatch:C3\nRoll Number:64\n");
 Stack st1, st2;
```

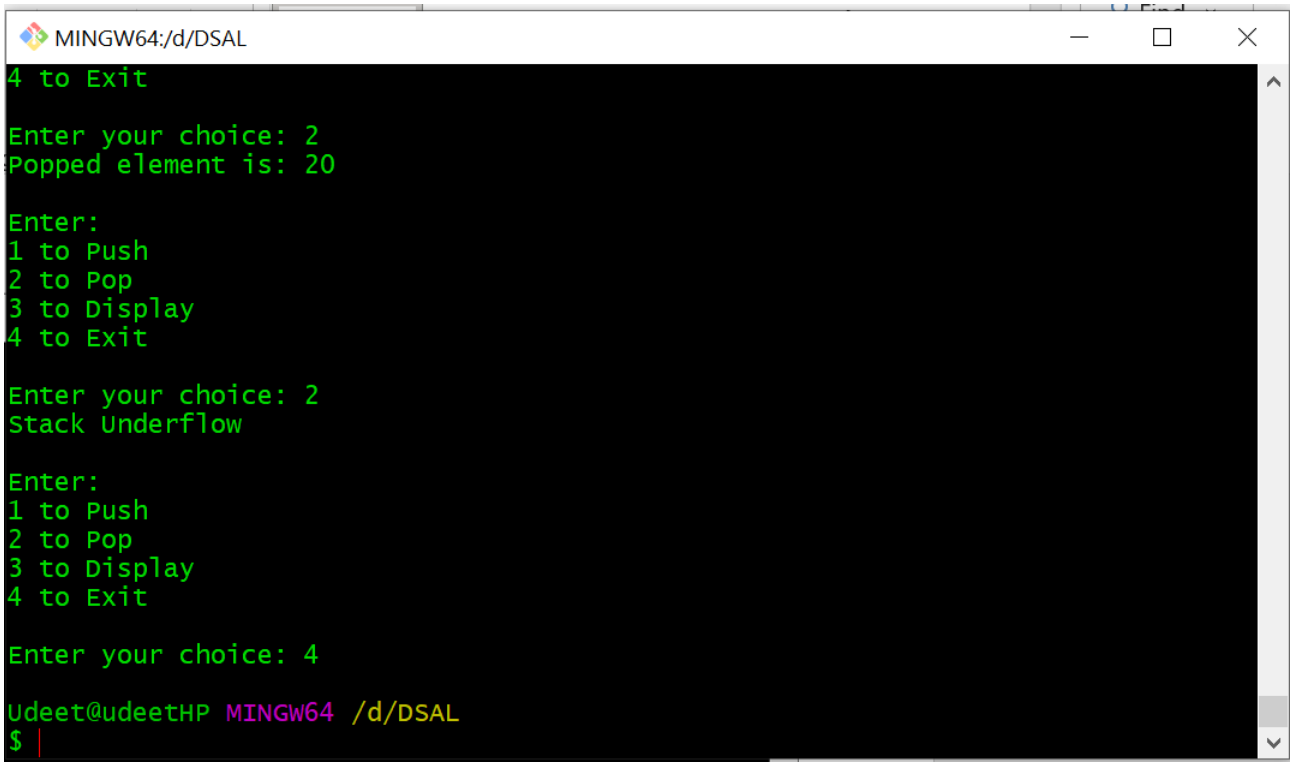
```
st1.top = st2.top = -1;
Stack *s1 = &st1;
Stack *s2 = &st2;
int ch, n;
int i = 0;
while (1)
{
printf("\nEnter:\n1 to Push\n2 to Pop\n3 to Display\n4 to Exit\n\nEnter your choice: ");
scanf("%d", &ch);
switch (ch)
{
case 1:
printf("\nEnter the element you want to push:\n");
scanf("%d", &n);
push(s1, n);
break;
case 2:
if (isEmpty(s2))
{
while (!isEmpty(s1))
{
push(s2, pop(s1));
}
n = pop(s2);
if (n != -1)
printf("Popped element is: %d\n", n);
```

```
else
printf("Stack Underflow\n");
}
else
{
n = pop(s2);
if (n != -1)
printf("Popped element is: %d\n", n);
else
printf("Stack Underflow\n");
}
break;
case 3:
for (int i = 0; i <=s1->top; i++)
{
printf("%d ", s1->arr[i]);
}
printf("\n");
break;
case 4:
exit(0);
}
}
return 0;
}
```


MINGW64:/d/DSAL
Udeet@udeetHP MINGW64 /d/DSAL
$ gcc q2.c
Udeet@udeetHP MINGW64 /d/DSAL
$ ./a
Name: Udeet Mittal
Batch:C3
Roll Number:64
Enter:
1 to Push
2 to Pop
3 to Display
4 to Exit
Enter your choice: 1
Enter the element you want to push:
10
Enter:
1 to Push
2 to Pop
3 to Display


MINGW64:/d/DSAL
4 to Exit
Enter your choice: 1
Enter the element you want to push:
20
Enter:
1 to Push
2 to Pop
3 to Display
4 to Exit
Enter your choice: 3
10 20
Enter:
1 to Push
2 to Pop
3 to Display
4 to Exit
Enter your choice: 2
Popped element is: 10


MINGW64:/d/DSAL
Enter your choice: 2
Popped element is: 10
Enter:
1 to Push
2 to Pop
3 to Display
4 to Exit
Enter your choice: 2
Popped element is: 20
Enter:
1 to Push
2 to Pop
3 to Display
4 to Exit
Enter your choice: 2
Stack Underflow
Enter:
1 to Push
2 to Pop


MINGW64:/d/DSAL
4 to Exit
Enter your choice: 2
Popped element is: 20
Enter:
1 to Push
2 to Pop
3 to Display
4 to Exit
Enter your choice: 2
Stack Underflow
Enter:
1 to Push
2 to Pop
3 to Display
4 to Exit
Enter your choice: 4
Udeet@udeetHP MINGW64 /d/DSAL
$