LAB – 5

SOLVED EXAMPLE :

1) Implement a queue of integers. Include functions insertq, deleteq and displayq.

CODE :

File name: 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]);
      }
    }
}
```

File name: queue.c

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

int main()
{
   printf("Name : Manoj M Mallya\nReg no : 200905130\nBatch : C2\n\n");
  queue q;
  q.front=-1;
  q.rear=-1;
  int ch,x,flag=1;
  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;
 }
```

OUTPUT :

```
Name : Manoj M Mallya
Reg no : 200905130
Batch : C2



1. Insert Queue
2. Delete Queue
3. Display Queue
4. Exit

Enter your choice: 1

Enter the Element:3


1. Insert Queue
2. Delete Queue
3. Display Queue
4. Exit

Enter your choice: 1

Enter the Element:6


1. Insert Queue
2. Delete Queue
3. Display Queue
4. Exit

Enter your choice: 1

Enter the Element:9
```

```
Enter the Element:9


1. Insert Queue
2. Delete Queue
3. Display Queue
4. Exit

Enter your choice: 3

Queue is:
3
6
9


1. Insert Queue
2. Delete Queue
3. Display Queue
4. Exit

Enter your choice: 2

Removed 3 from the Queue


1. Insert Queue
2. Delete Queue
3. Display Queue
4. Exit

Enter your choice: 3
```

```
Enter your choice: 3

Queue is:
6
9


1. Insert Queue
2. Delete Queue
3. Display Queue
4. Exit

Enter your choice: 4
```

EXERCISE :

1) Implement a circular queue of Strings using structures. Include functions insertcq, deletecq and displaycq.

CODE :

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define max_size 10
#define max_str 20
typedef struct{
   char **arr;
   int front,rear;
}QUE;
void initialize(QUE *cq){
   int i;
   cq->front = -1;
   cq->rear = -1;
   cq->arr = malloc(sizeof(char*)*max_size);
   for(i=0;i<max_size;i++){
      cq->arr[i] = malloc(sizeof(char)*max_str);
   }
}
void insertcq(QUE *cq,char *str){
   if(cq->front == cq->rear && cq->rear == -1){
      cq->rear=cq->front=0;
      strcpy(cq->arr[cq->rear],str);
      return;
   }
   if(cq->front == ((cq->rear)+1)%max_size){
      printf("Queue is full\n");
      return;
   }
   cq->rear = ((cq->rear)+1)%max_size;
   strcpy(cq->arr[cq->rear],str);
}
void deletecq(QUE *cq){
   char *ele;
   if(cq->front == cq->rear){
      printf("Queue underflow\n");
      return;
```

```
    }
    else{
        ele = cq->arr[cq->front];
        printf("Deleted string: %s\n", ele);
        cq->front=((cq->front)+1)%max_size;
    }
}
void display(QUE *cq){
    int i;
    if(cq->rear == cq->front){
        printf("Queue is empty\n");
        return;
    }
    else{
        for(i=cq->front;i!=cq->rear;i=(i+1)%max_size){
            printf("%s ",cq->arr[i]);
        }
        printf("%s\n", cq->arr[i]);
    }
}
int main(){
    printf("Name : Manoj M Mallya\nReg no : 200905130\nBatch : C2\n\n");
    QUE cq;
    initialize(&cq);
    int ch;
    char x[max_str];
    do{
        printf("\n1.Insert\n2.Delete\n3.Display\n4.Exit\n");
        printf("Enter your choice\n");
        scanf("%d",&ch);
        switch(ch)
        {
            case 1:
                printf("Enter a string\n");
                scanf("%s",x);
                insertcq(&cq,x);
                break;
            case 2:
                deletecq(&cq);
                break;
            case 3:
                display(&cq);
```

```
            break;
        case 4:
            exit(5);
      }
   }while(ch!=4);
   return 0;
}
```

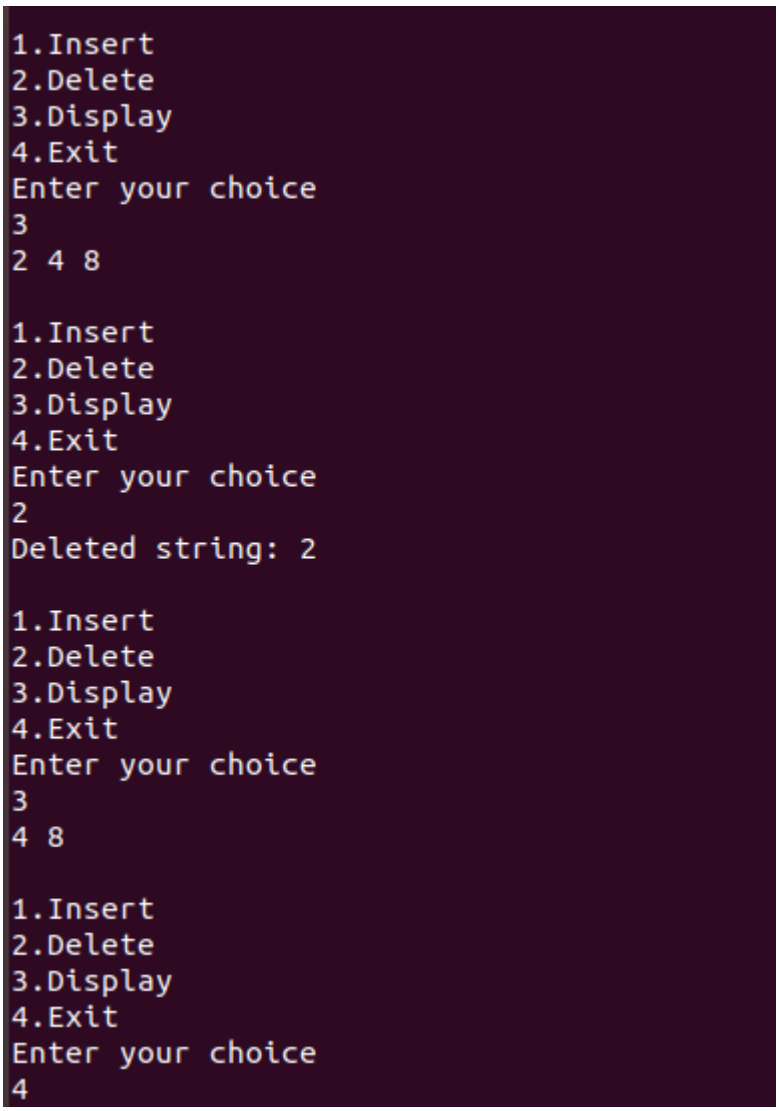
OUTPUT :

```
Student@project-lab:~/Desktop/200905130/DSAlab5$ gcc lab5_1.c
Student@project-lab:~/Desktop/200905130/DSAlab5$ ./a.out
Name : Manoj M Mallya
Reg no : 200905130
Batch : C2


1.Insert
2.Delete
3.Display
4.Exit
Enter your choice
1
Enter a string
2

1.Insert
2.Delete
3.Display
4.Exit
Enter your choice
1
Enter a string
4

1.Insert
2.Delete
3.Display
4.Exit
Enter your choice
1
Enter a string
8
```



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).

CODE :

```
#include <stdio.h>

#include <stdlib.h>

#define MAX 5

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)--];
}
int main() {
Stack s1, s2;
s1.top = s2.top = -1;
int ch,n;
int i=0;
while (1){
printf("Enter:\n1 to Push\n2 to Pop\n3 to Display\n4 to Exit\nEnter your choice : ");
```

```
scanf("%d",&ch);
switch(ch){
case 1 :
printf("Enter the element you want to push : ");
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 : %d\n",n);
else
printf("Underflow\n");
}
else {
n=pop(&s2);
if(n!=-1)
printf("Popped : %d\n",n);
else
printf("Underflow\n");
```

```
}
break;
case 3:
for(int i=0; i<MAX; i++){
printf(" %d", s1.arr[i]);
}
printf("\n");
break;
case 4:
exit(0);
}
}
return 0;
}
```

OUTPUT :

```
Name : Manoj M Mallya
Reg no : 200905130
Batch : C2

Enter:
1 to Push
2 to Pop
3 to Display
4 to Exit
Enter your choice : 1
Enter the element you want to push : 3
Enter:
1 to Push
2 to Pop
3 to Display
4 to Exit
Enter your choice : 1
Enter the element you want to push : 2
Enter:
1 to Push
2 to Pop
3 to Display
4 to Exit
Enter your choice : 1
Enter the element you want to push : 3
Enter:
1 to Push
2 to Pop
3 to Display
4 to Exit
Enter your choice : 3
 3 2 3
```

```
Enter:
1 to Push
2 to Pop
3 to Display
4 to Exit
Enter your choice : 2
Popped : 3
Enter:
1 to Push
2 to Pop
3 to Display
4 to Exit
Enter your choice : 2
Popped : 2
Enter:
1 to Push
2 to Pop
3 to Display
4 to Exit
Enter your choice : 4
```

****