main.c

```
#include <stdio.h>

int MAXSIZE = 8;
int stack[8];
int top = -1;

int isempty() {

   if(top == -1)
      return 1;
   else
      return 0;
}

int isfull() {

   if(top == MAXSIZE)
      return 1;
   else
      return 0;
}

int peek() {
   return stack[top];
}

int pop() {
   int data;
```

main.c

```
   if(!isempty()) {
      data = stack[top];
      top = top - 1;
      return data;
   } else {
      printf("Could not retrieve data, Stack is empty.\n");
   }
}

int push(int data) {

   if(!isfull()) {
      top = top + 1;
      stack[top] = data;
   } else {
      printf("Could not insert data, Stack is full.\n");
   }
}

int main() {
   push(3);
   push(5);
   push(9);
   push(1);
   push(12);
   push(15);

   printf("Element at top of the stack: %d\n" ,peek());
   printf("Elements: \n");
   while(!isempty()) {
```

input

```
    printf("Element at top of the stack: %d\n" ,peek());
    printf("Elements: \n");
    while(!isempty()) {
        int data = pop();
        printf("%d\n",data);
    }

    printf("Stack full: %s\n" , isfull()?"true":"false");
    printf("Stack empty: %s\n" , isempty()?"true":"false");

    return 0;
}
```

input

input

```
Element at top of the stack: 15
Elements:
15
12
1
9
5
3
Stack full: false
Stack empty: true


...Program finished with exit code 0
Press ENTER to exit console.
```

main.c

```c
#include <stdio.h>
#include<stdlib.h>
#define MAX 50
void insert();
void delete();
void display();
int queue_array[MAX];
int rear = - 1;
int front = - 1;
int main()
{
int choice;
while (1)
{
printf("1.Insert element to queue \n");
printf("2.Delete element from queue \n");
printf("3.Display all elements of queue \n");
printf("4.Quit \n");
printf("Enter your choice : ");
scanf("%d", &choice);
switch(choice)
{
case 1:
insert();
break;
case 2:
delete();
break;
case 3:
display();
```

main.c

```c
display();
break;
case 4:
exit(1);
default:
printf("Wrong choice \n");
}
}
}
void insert()
{
int item;
if(rear == MAX - 1)
printf("Queue Overflow \n");
else
{
if(front== - 1)
front = 0;
printf("Inset the element in queue : ");
scanf("%d", &item);
rear = rear + 1;
queue_array[rear] = item;
}
}
void delete()
{
if(front == - 1 || front > rear)
{
printf("Queue Underflow \n");
return;
```

```
66 return;
67 }
68 else
69 {
70 printf("Element deleted from queue is : %d\n", queue_array[front]);
71 front = front + 1;
72 }
73 }
74 void display()
75 {
76 int i;
77 if(front == - 1)
78 printf("Queue is empty \n");
79 else
80 {
81 printf("Queue is : \n");
82 for(i = front; i <= rear; i++)
83 printf("%d ", queue_array[i]);
84 printf("\n");
85 }
86 }
87
```

input

input

```
1.Insert element to queue
2.Delete element from queue
3.Display all elements of queue
4.Quit
Enter your choice : 1
Inset the element in queue : 3
1.Insert element to queue
2.Delete element from queue
3.Display all elements of queue
4.Quit
Enter your choice : 1
Inset the element in queue : 4
1.Insert element to queue
2.Delete element from queue
3.Display all elements of queue
4.Quit
Enter your choice : 3
Queue is :
3 4
1.Insert element to queue
2.Delete element from queue
3.Display all elements of queue
4.Quit
Enter your choice : 2
Element deleted from queue is : 3
1.Insert element to queue
2.Delete element from queue
3.Display all elements of queue
4.Quit
```