

WAP to simulate the working of a circular queue of integers using an array. Provide the following operations: Insert, Delete & Display. The program should print appropriate messages for queue empty and queue overflow conditions.

Program:

```
#include<stdio.h>
#include<string.h>
#include<conio.h>
#define SIZE 5

int queue[SIZE];
int front = -1, rear = -1;
void enqueue(int value);
int dequeue();
void display();

int main(){
    int value, option;
    while(1){
        printf("Enter 1 to INSERT, 2 to DELETE, 3 to DISPLAY, 4 to EXIT: \n");
        scanf("%d",&option);
        switch(option){
            case 1:
                printf("Enter value to be inserted: \n");
                scanf("%d",&value);
                enqueue(value);
                break;
            case 2:
                value = dequeue();
                printf("Value deleted is: %d\n",value);
                break;
            case 3:
                display();
                break;
            case 4:
                exit(1);
                break;
            default:
                printf("Invalid Input.\n");
        }
    }
}

void enqueue(int value){
    if(front==(rear+1)%SIZE){
        printf("Queue OVERFLOW\n");
    }else if (front== -1 && rear== -1){
        front++;
    }
```

```

        rear++;
        queue[rear] = value;
    }else {
        rear = (rear+1)%SIZE;
        queue[rear] = value;
    }
}

int dequeue(){
    int value;
    if((front==-1 && rear==-1)|| front>rear){
        printf("Queue UNDERFLOW\n");
    }else if(front==rear){
        value = queue[front];
        front=rear=-1;
    }else{
        value = queue[front];
        front = (front+1)%SIZE;
    }
    return value;
}

void display(){
    int i;
    if(front==-1){
        printf("Queue is EMPTY.\n");
    }else{
        printf("Queue elements are: \n");
        for(i=front;i!=rear;i=((i+1)%SIZE))
        {
            printf("%d\n", queue[i]);
        }
        printf("%d\n",queue[i]);
    }
}
}

```



OUTPUT:

```
Enter 1 to INSERT, 2 to DELETE, 3 to DISPLAY, 4 to EXIT:
1
Enter value to be inserted:
12
Enter 1 to INSERT, 2 to DELETE, 3 to DISPLAY, 4 to EXIT:
1
Enter value to be inserted:
23
Enter 1 to INSERT, 2 to DELETE, 3 to DISPLAY, 4 to EXIT:
3
Queue elements are:
12
23
Enter 1 to INSERT, 2 to DELETE, 3 to DISPLAY, 4 to EXIT:
2
Value deleted is: 12
Enter 1 to INSERT, 2 to DELETE, 3 to DISPLAY, 4 to EXIT:
2
Value deleted is: 23
Enter 1 to INSERT, 2 to DELETE, 3 to DISPLAY, 4 to EXIT:
2
Queue UNDERFLOW
Value deleted is: 4214942
Enter 1 to INSERT, 2 to DELETE, 3 to DISPLAY, 4 to EXIT:
1
Enter value to be inserted:
12
```

```
Enter 1 to INSERT, 2 to DELETE, 3 to DISPLAY, 4 to EXIT:
1
Enter value to be inserted:
23
Enter 1 to INSERT, 2 to DELETE, 3 to DISPLAY, 4 to EXIT:
1
Enter value to be inserted:
34
Enter 1 to INSERT, 2 to DELETE, 3 to DISPLAY, 4 to EXIT:
1
Enter value to be inserted:
45
Enter 1 to INSERT, 2 to DELETE, 3 to DISPLAY, 4 to EXIT:
1
Enter value to be inserted:
56
Enter 1 to INSERT, 2 to DELETE, 3 to DISPLAY, 4 to EXIT:
1
Enter value to be inserted:
67
Queue OVERFLOW
Enter 1 to INSERT, 2 to DELETE, 3 to DISPLAY, 4 to EXIT:
4
```

```
Process returned 1 (0x1)   execution time : 76.081 s
Press any key to continue.
```