Experiment No:3 [Circular Queue]

Name: Shubham Shatrughna Shendage
Class:SE-IT
Batch:D
Roll_No:561

```
#include <iostream>
using namespace std;
#define SIZE 5
int A[SIZE];
int front=-1;
int rear=-1;

bool isempty()
{
        if(front==-1&&rear==-1)

                return true;
        else
                return false;

}
void enqueue(int value)
{
        if((rear+1)%SIZE==front)
                        cout<<"Queue is full \n";
        else
        {
                if(front==-1)
                        front=0;
                rear=(rear+1)%SIZE;
                A[rear]=value;
        }

}
void dequeue()
{
        if(isempty())
                cout<<"Queue is empty";
        else
                if(front==rear)
                        front=rear=-1;
                else
                        front=(front+1)%SIZE;
}
void showfront()
{
        if(isempty())
                cout<<"Queue is empty\n";
        else
                cout<<"element at front is:"<<A[front];
}
void displayQueue()
{
        if(isempty())
                cout<<"Queue is empty\n";
        else
        {
                int i;
                if(front<=rear)
                {
                        for(i=front;i<=rear;i++)
                                cout<<A[i]<<" ";
                }
                else
                {
                        i=front;
                        while(i<SIZE)
                        {
                                cout<<A[i]<<" ";
                                i++;
                        }
                        i=0;
                        while(i<=rear)
                        {
                                cout<<A[i]<<" ";
                                i++;
                        }
                }
        }
}
int main()
{
        int choice,flag=1,value;
        while(flag==1)
        {
                cout<<"\n1.enqueue 2.dequeue 3.showfront 4.displayQueue 5,exit \n";
                cin>>choice;
                switch(choice)
                {
                case 1:cout<<"Enter value:\n";
                cin>>value;
                enqueue(value);
                break;
                case 2:dequeue();
                break;
                case 3:showfront();
                break;
                case 4:displayQueue();
                break;
                case 5:flag=0;
                cout<<"code exit";
                break;

                }
        }
        return 0;
}
```

Output:

```
1.enqueue 2.dequeue 3.showfront 4.displayQueue 5,exit
1
Enter value:
11

1.enqueue 2.dequeue 3.showfront 4.displayQueue 5,exit
1
Enter value:
22

1.enqueue 2.dequeue 3.showfront 4.displayQueue 5,exit
1
Enter value:
33

1.enqueue 2.dequeue 3.showfront 4.displayQueue 5,exit
1
Enter value:
44

1.enqueue 2.dequeue 3.showfront 4.displayQueue 5,exit
1
Enter value:
55

1.enqueue 2.dequeue 3.showfront 4.displayQueue 5,exit
1
Enter value:
66
Queue is full

1.enqueue 2.dequeue 3.showfront 4.displayQueue 5,exit
4
11 22 33 44 55
1.enqueue 2.dequeue 3.showfront 4.displayQueue 5,exit
3
element at front is:11
1.enqueue 2.dequeue 3.showfront 4.displayQueue 5,exit
2

1.enqueue 2.dequeue 3.showfront 4.displayQueue 5,exit
4
22 33 44 55
1.enqueue 2.dequeue 3.showfront 4.displayQueue 5,exit
2

1.enqueue 2.dequeue 3.showfront 4.displayQueue 5,exit
4
33 44 55
1.enqueue 2.dequeue 3.showfront 4.displayQueue 5,exit
2

1.enqueue 2.dequeue 3.showfront 4.displayQueue 5,exit
4
44 55
1.enqueue 2.dequeue 3.showfront 4.displayQueue 5,exit
2

1.enqueue 2.dequeue 3.showfront 4.displayQueue 5,exit
```

```
4
55
1.enqueue 2.dequeue 3.showfront 4.displayQueue 5,exit
2

1.enqueue 2.dequeue 3.showfront 4.displayQueue 5,exit
4
Queue is empty

1.enqueue 2.dequeue 3.showfront 4.displayQueue 5,exit
3
Queue is empty

1.enqueue 2.dequeue 3.showfront 4.displayQueue 5,exit
5
code exit
```