# ADDITIONAL EXPERIMENT

Harsh Kasliwal
2003085
C21

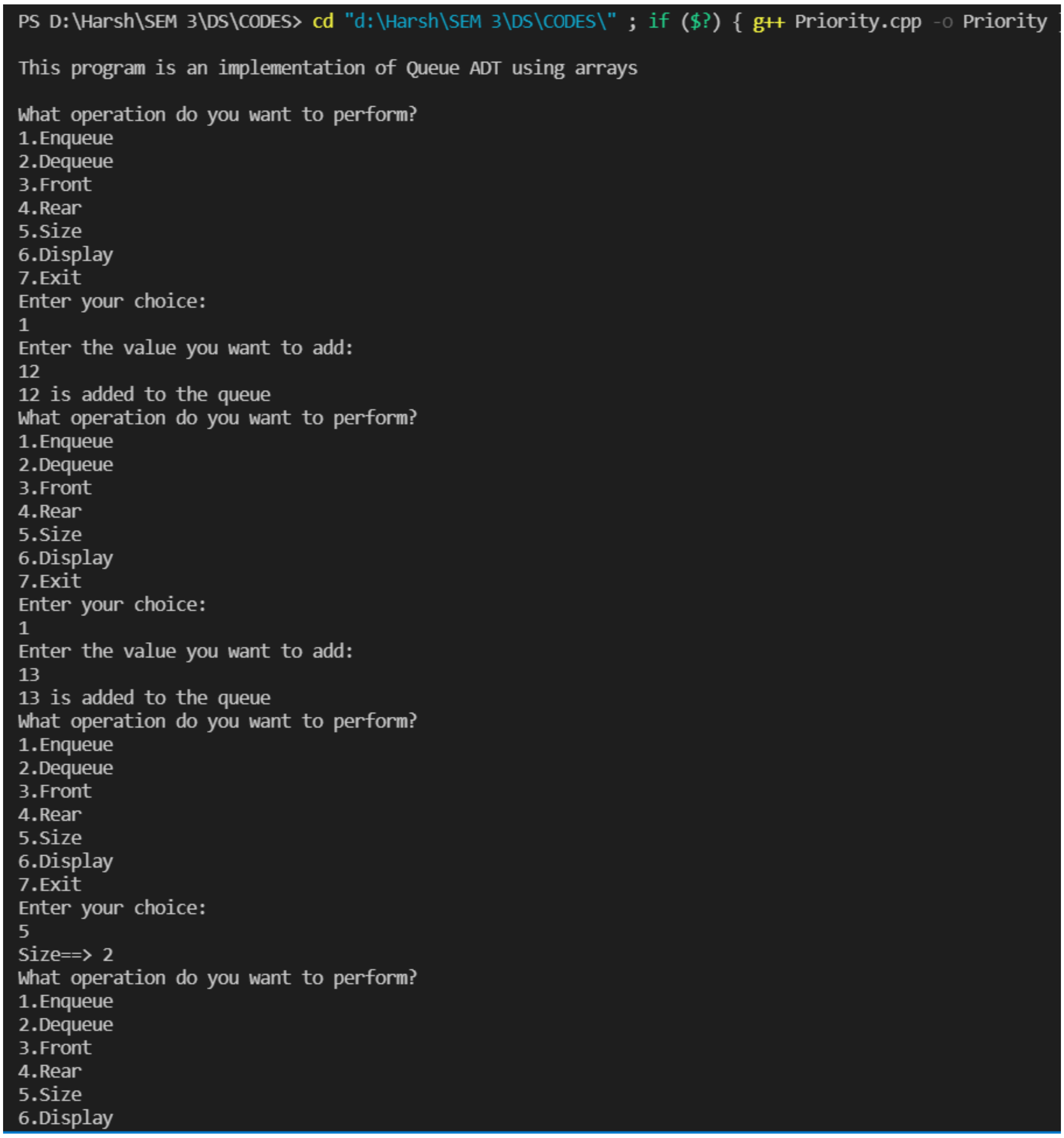
**Program:** Write a menu driven code to implement Priority Queue using Arrays.

```
#include <iostream>
using namespace std;

void enqueue();
void dequeue();
void peekfront();
void peekrear();
void Size();
void display();

#define max 5
int queue[max], choice, front = -1, rear = -1, value;

int main()
{

    cout<<"\nThis program is an implementation of Queue ADT using arrays\n";

    while (choice != 7)
    {
        cout<<"\nWhat operation do you want to perform?\n";
        cout<<"1.Enqueue\n";
        cout<<"2.Dequeue\n";
        cout<<"3.Front\n";
        cout<<"4.Rear\n";
        cout<<"5.Size\n";
        cout<<"6.Display\n";
        cout<<"7.Exit\n";
        cout<<"Enter your choice:"<<endl;
        cin>>choice;

        switch (choice)
        {
        case 1:
            enqueue();
            break;
```

```cpp
        case 2:
            dequeue();
            break;

        case 3:
            peekfront();
            break;

        case 4:
            peekrear();
            break;

        case 5:
            Size();
            break;

        case 6:
            display();
            break;

        case 7:
            break;
        }
    }

    return 0;
}

void dequeue()
{
    int value;
    if ((front == -1) && (rear == -1))
    {
        cout<<"\nQueue is empty\n";
    }
    else
    {
        value = queue[front];
        if (front == rear)
        {
            front = -1;
            rear = -1;
        }
        else
        {
            int pos = 0;
            int p = queue[0];
            for (int i = 1; i <= rear; i++)
            {
                if (queue[i] > p)
                {
                    pos = i;
                    p = queue[i];
                }
            }
            value = p;
            for (int i = pos; i < rear; i++)
            {
                queue[i] = queue[i + 1];
            }
            rear--;
        }
        cout<<value<< " is deleted from the queue";
    }
}

void enqueue()
{

    int value;
    if (rear == max - 1)
    {
        cout<<"\nQueue is full\n";
    }
    else
    {
        cout<<"Enter the value you want to add:"<<endl;
        cin>>value;
        if ((front == -1) && (rear == -1))
        {
            front = 0;
            rear = 0;
            queue[rear] = value;
           cout<<value << " is added to the queue";
        }
        else
        {
            rear++;
            queue[rear] = value;
            cout<<value << " is added to the queue";
        }
    }
}

void peekfront()
```

```
{
    if ((front == -1) && (rear == -1))
    {
        cout<<"\nUNDERFLOW\n";
    }
    else
    {
        cout<<queue[front]<< " is at the front end\n";
    }
}

void peekrear()
{
    if ((front == -1) && (rear == -1))
    {
        cout<<"\nUNDERFLOW\n";
    }
    else
    {
        cout<<queue[rear]<< " is at the rear end\n";
    }
}

void Size()
{
    cout<<"Size==> "<<(rear - front) + 1;
}

void display()
{
    if ((front == -1) && (rear == -1))
    {
        cout<<"\nUNDERFLOW\n";
    }
    else
    {
        cout<<"\nThe elements in the queue are:";
        for (int i = front; i <= rear; i++)
        {
            cout<<" "<<queue[i];
        }
        cout<<"\n";
    }
}
```

## OUTPUT:



```
PS D:\Harsh\SEM 3\DS\CODES> cd "d:\Harsh\SEM 3\DS\CODES\" ; if ($?) { g++ Priority.cpp -o Priority

This program is an implementation of Queue ADT using arrays

What operation do you want to perform?
1.Enqueue
2.Dequeue
3.Front
4.Rear
5.Size
6.Display
7.Exit
Enter your choice:
1
Enter the value you want to add:
12
12 is added to the queue
What operation do you want to perform?
1.Enqueue
2.Dequeue
3.Front
4.Rear
5.Size
6.Display
7.Exit
Enter your choice:
1
Enter the value you want to add:
13
13 is added to the queue
What operation do you want to perform?
1.Enqueue
2.Dequeue
3.Front
4.Rear
5.Size
6.Display
7.Exit
Enter your choice:
5
Size==> 2
What operation do you want to perform?
1.Enqueue
2.Dequeue
3.Front
4.Rear
5.Size
6.Display
```

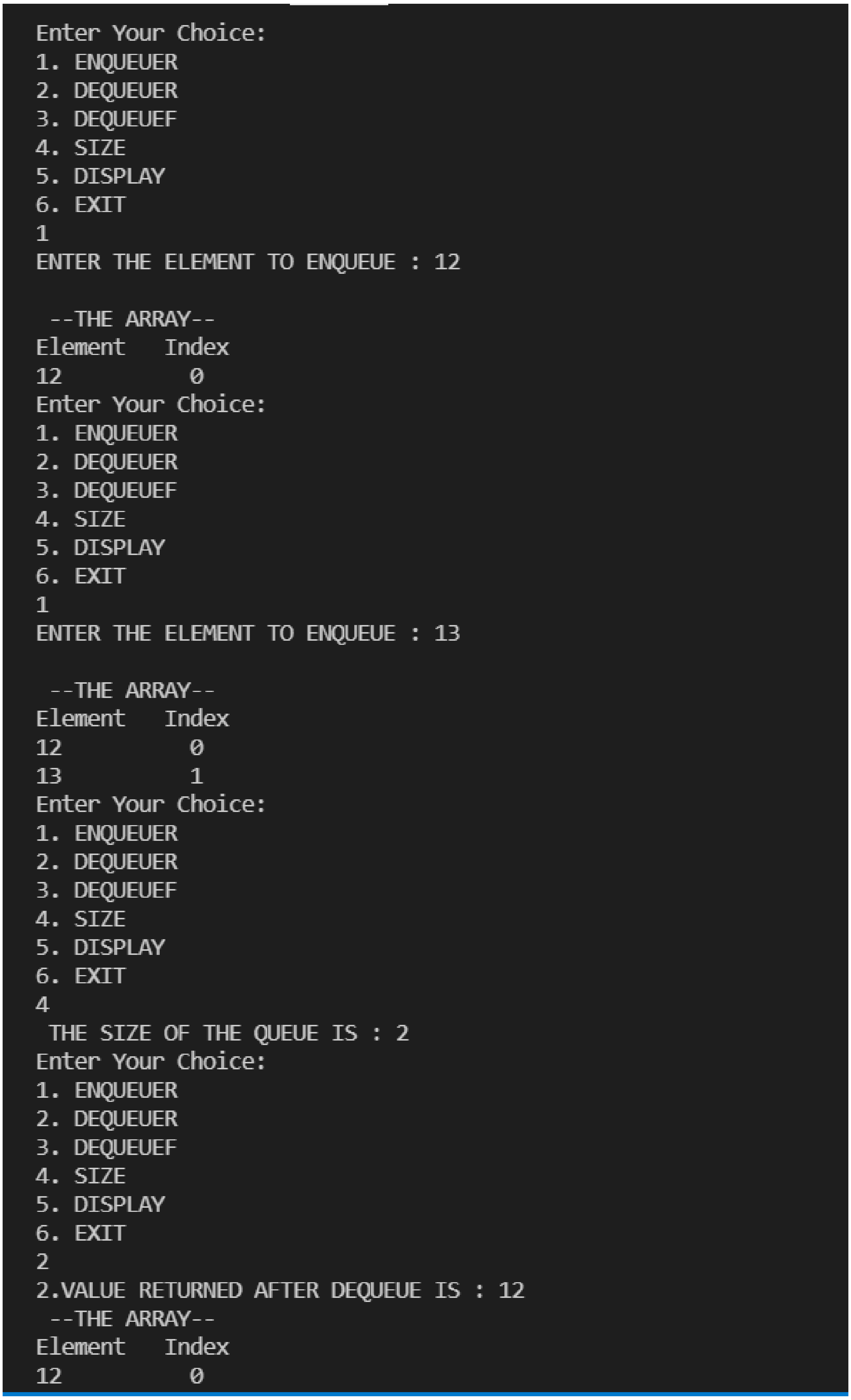
**Program 2**: Write a menu driven code to implement Double Ended Queue using Arrays.

```cpp
#include <iostream>
using namespace std;
#include <conio.h>
int Max = 4;
int arr[4];
int front = -1;
int rear = -1;
int count = 0;

int isEmpty()
{
    if (front == -1 && rear == -1)
    {
        return 1;
    }
    else
    {
        return 0;
    }
}
int isFull()
{
    if ((front == 0 && rear == Max - 1) || (front == rear + 1))
    {
        return 1;
    }
    else
    {
        return 0;
    }
}

void display()
{
    cout<<"\n --THE ARRAY-- ";
    cout<<"\nElement   Index";
    int ch = 0;
    if (isEmpty() == 1)
    {
        cout<<"\n --EMPTY QUEUE-- ";
    }
    else if (front <= rear)
    {
        for (int i = front; i <= rear; i++)
        {
            cout<<"\n"<<arr[i]<<"          "<<i;
        }
    }
    else
    {
        for (int i = front; i <= Max - 1; i++)
        {
            cout<<"\n"<<arr[i]<<"          "<<i;
        }
        for (int i = 0; i <= rear; i++)
        {

            cout<<"\n"<<arr[i]<<"          "<<i;
        }
    }
}

void EnqueueF()
{

    int flag = 0;
    int enq;
    if (isFull() == 1)
    {
        cout<<" --OVERFLOW-- ";
        flag = 1;
    }
    else if ((front == -1) && (rear == -1))
    {
        front = 0;
        rear = 0;
    }
    else if (front == 0)
    {
        front = Max - 1;
    }
    else
    {
        front = front - 1;
    }
    if (flag == 0)
    {
        cout<<"ENTER THE ELEMENT TO ENQUEUE : ";
        count++;
        cin>>enq;
        arr[front] = enq;
    }
    display();
```

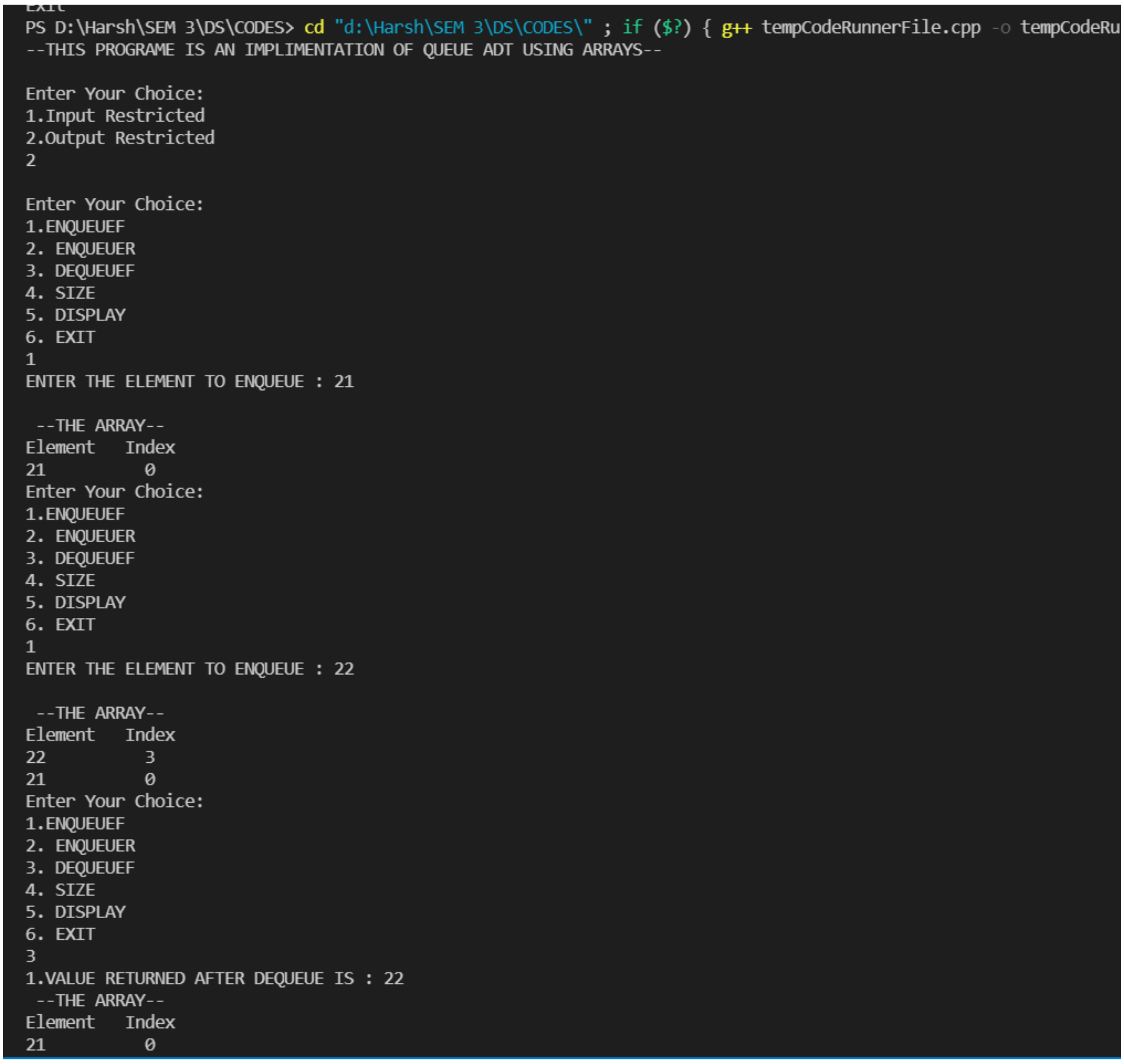
```
}

void EnqueueR()
{

    int flag = 0;
    int enq;
    if (isFull() == 1)
    {
        cout<<" --OVERFLOW-- ";
        flag = 1;
    }
    else if ((front == -1) && (rear == -1))
    {
        front = 0;
        rear = 0;
    }
    else if (rear == (Max - 1))
    {
        rear = 0;
    }
    else
    {
        rear = rear + 1;
    }
    if (flag == 0)
    {
        cout<<"ENTER THE ELEMENT TO ENQUEUE : ";
        count++;
        cin>>enq;
        arr[rear] = enq;
    }
    display();
}

void DequeueF()
{
    if (isEmpty() == 1)
    {
      cout<<" --UNDERFLOW-- ";
    }
    else
    {
        int value = arr[front];
        count--;
        cout<<"1.VALUE RETURNED AFTER DEQUEUE IS : "<<value;
        if (front == rear)
        {
            front = -1;
            rear = -1;
        }
        else if (front == Max - 1)
        {
            front = 0;
        }
        else
        {
            front = front + 1;
        }
    }
    display();
}

void DequeueR()
{
    if (isEmpty() == 1)
    {
        cout<<" --UNDERFLOW-- ";
    }
    else
    {
        int value = arr[front];
        count--;
        cout<<"2.VALUE RETURNED AFTER DEQUEUE IS : "<<value;
    }
    if (front == rear)
    {
        front = -1;
        rear = -1;
    }
    else if (rear == 0)
    {
        rear = Max - 1;
    }
    else
    {
        rear = rear - 1;
    }
    display();
}

void size()
{
    if (isEmpty() == 1)
    {
        cout<<"\n --EMPTY QUEUE-- ";
```

```cpp
    }
    else
    {
        cout<<" THE SIZE OF THE QUEUE IS : "<<count;
    }
}

int main()
{
    int temp;
    int n;
    int ch = 0;
    int sh = 0;
    cout<<"--THIS PROGRAME IS AN IMPLIMENTATION OF QUEUE ADT USING ARRAYS--
\n";
    cout<<"\nEnter Your Choice: ";
    cout<<"\n1.Input Restricted\n2.Output Restricted\n";
    cin>>sh;

    while (ch != 6)
    {
        if (sh == 1)
        {

            cout<<"\nEnter Your Choice: ";
            cout<<"\n1. ENQUEUER \n2. DEQUEUER \n3. DEQUEUEF\n4. SIZE\n5.
DISPLAY\n6. EXIT\n";
            cin>>ch;
            switch (ch)
            {
            case 1:
                EnqueueR();
                break;
            case 2:
                DequeueR();
                break;
            case 3:
                DequeueF();
                break;
            case 4:
                size();
                break;
            case 5:
                display();
                break;
            case 6:
                cout<<"Exit"<<endl;
                break;
            default:
                cout<<"Invalid Choice";
                break;
            }
        }
        if (sh == 2)
        {
            cout<<"\nEnter Your Choice:";
            cout<<"\n1.ENQUEUEF\n2. ENQUEUER \n3. DEQUEUEF\n4. SIZE\n5.
DISPLAY\n6. EXIT\n";
            cin>>ch;
            switch (ch)
            {

            case 1:
                EnqueueF();
                break;
            case 2:
                EnqueueR();
                break;
            case 3:
                DequeueF();
                break;
            case 4:
                size();
                break;
            case 5:
                display();
                break;
            case 6:
                cout<<"Exit"<<endl;
                break;
            default:
                cout<<"Invalid Choice";
                break;
            }
        }
    }
}
```

**OUTPUT:**



```
Enter Your Choice:
1. ENQUEUER
2. DEQUEUER
3. DEQUEUEF
4. SIZE
5. DISPLAY
6. EXIT
1
ENTER THE ELEMENT TO ENQUEUE : 12

 --THE ARRAY--
Element   Index
12          0
Enter Your Choice:
1. ENQUEUER
2. DEQUEUER
3. DEQUEUEF
4. SIZE
5. DISPLAY
6. EXIT
1
ENTER THE ELEMENT TO ENQUEUE : 13

 --THE ARRAY--
Element   Index
12          0
13          1
Enter Your Choice:
1. ENQUEUER
2. DEQUEUER
3. DEQUEUEF
4. SIZE
5. DISPLAY
6. EXIT
4
 THE SIZE OF THE QUEUE IS : 2
Enter Your Choice:
1. ENQUEUER
2. DEQUEUER
3. DEQUEUEF
4. SIZE
5. DISPLAY
6. EXIT
2
2.VALUE RETURNED AFTER DEQUEUE IS : 12
 --THE ARRAY--
Element   Index
12          0
```



```
PS D:\Harsh\SEM 3\DS\CODES> cd "d:\Harsh\SEM 3\DS\CODES\" ; if ($?) { g++ tempCodeRunnerFile.cpp -o tempCodeRu
--THIS PROGRAME IS AN IMPLIMENTATION OF QUEUE ADT USING ARRAYS--

Enter Your Choice:
1.Input Restricted
2.Output Restricted
2

Enter Your Choice:
1.ENQUEUEF
2. ENQUEUER
3. DEQUEUEF
4. SIZE
5. DISPLAY
6. EXIT
1
ENTER THE ELEMENT TO ENQUEUE : 21

 --THE ARRAY--
Element   Index
21          0
Enter Your Choice:
1.ENQUEUEF
2. ENQUEUER
3. DEQUEUEF
4. SIZE
5. DISPLAY
6. EXIT
1
ENTER THE ELEMENT TO ENQUEUE : 22

 --THE ARRAY--
Element   Index
22          3
21          0
Enter Your Choice:
1.ENQUEUEF
2. ENQUEUER
3. DEQUEUEF
4. SIZE
5. DISPLAY
6. EXIT
3
1.VALUE RETURNED AFTER DEQUEUE IS : 22
 --THE ARRAY--
Element   Index
21          0
```