

# LAB 03

M.Huzaifa Mustafa

SP22-BSCS-0046

1. Implement Queue.

SOURCE CODE:

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

```
class Queue{
private:
    int front;
    int rare;
    int arr_queue[];
    int queuesize;

public:
    Queue(){
        front = -1;
        rare = -1;
    }
    Queue(int queuesize){
        arr_queue[queuesize];
    }
    void enqueue(int value){
        if(front == -1){
            front = 0;
        }
        if(rare < queuesize-1){
            arr_queue[++rare] = value;
        }
    }
    void dequeue(){
        if(front != -1){
            cout<<arr_queue[front++];
        }
        else{
            cout<<"Queue is Empty : "<<endl;
        }
    }
    void peek(){
        if(front != -1){
```

```

        cout<<arr_queue[front];
    }
    cout<<"Queue is Empty : "<<endl;
}

};

int main()
{
    Queue obj;

    int option;
    int value;

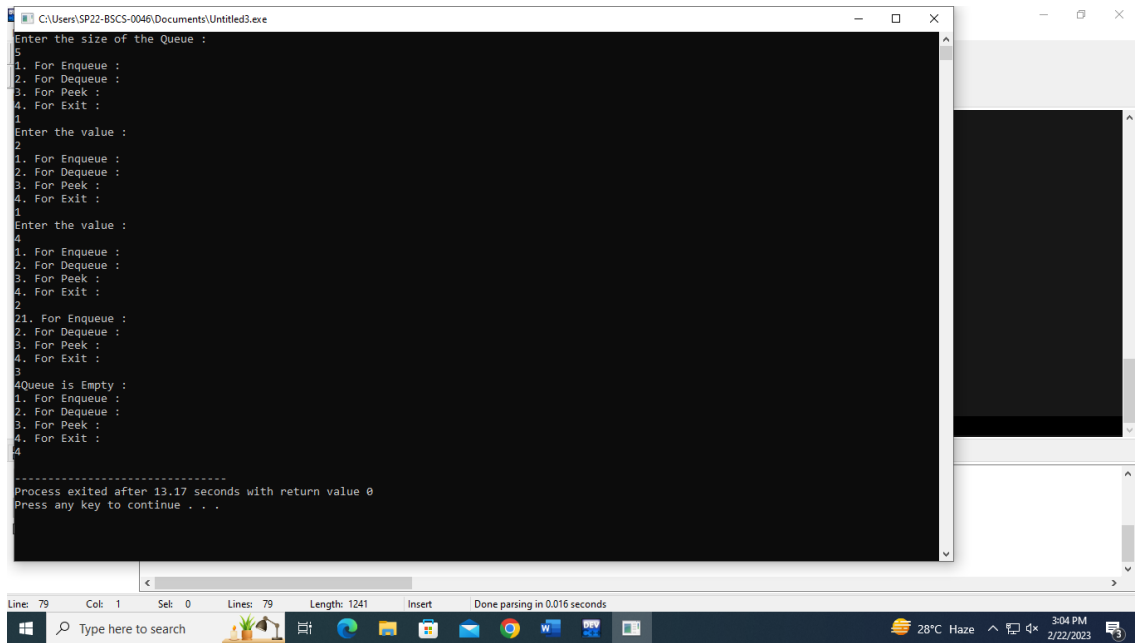
    cout<<"Enter the size of the Queue : "<<endl;
    cin>>value;

    do{
        cout<<"1. For Enqueue : "<<endl;
        cout<<"2. For Dequeue : "<<endl;
        cout<<"3. For Peek : "<<endl;
        cout<<"4. For Exit : "<<endl;
        cin>>option;
        if(option == 1){
            cout<<"Enter the value : "<<endl;
            cin>>value;
            obj.enqueue(value);
        }
        else if(option == 2){
            obj.dequeue();
        }
        else if(option == 3){
            obj.peek();
        }
        else{
            option = 0;
        }
    }while(option != 0);

    return 0;
}

```

PICTURE:



```
C:\Users\SP22-BSCS-0046\Documents\Untitled3.exe
Enter the size of the Queue :
5
1. For Enqueue :
2. For Dequeue :
3. For Peek :
4. For Exit :
1
Enter the value :
2
1. For Enqueue :
2. For Dequeue :
3. For Peek :
4. For Exit :
1
Enter the value :
4
1. For Enqueue :
2. For Dequeue :
3. For Peek :
4. For Exit :
2
21. For Enqueue :
2. For Dequeue :
3. For Peek :
4. For Exit :
3
4Queue is Empty :
1. For Enqueue :
2. For Dequeue :
3. For Peek :
4. For Exit :
4
-----
Process exited after 13.17 seconds with return value 0
Press any key to continue . . .
```

## 2. Implement Circular Queue.

SOURCE CODE:

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

```
//Implement Queue
```

```
class Queue{
private:
    int front,rear;
    int queuesize;
    int arr_queue[];

public:
    Queue(){
        front = -1;
        rear = -1;
        cout<<"Enter Queue Size: ";
        cin>>queuesize;
        arr_queue[queuesize];
    }
    void enqueue(int value){
        if(front == -1){
            front = 0;
        }
        if(rear < queuesize - 1){
            rear =(rear+1)%queuesize;
```

```

        arr_queue[rear] = value;
    }
}
}
void dequeue(){
    if(!front != -1 && front<queuesize){
        cout<<arr_queue[front]<<endl;
        front = (front+1)%queuesize;
    }
}
void peek(){
    if(front != -1){
        cout<<arr_queue[front];
    }
}
};

int main(){
    Queue Q1;
    int qsize;
    int option;
    int value;

    do{
        cout<<endl;
        cout<<"1 for Enqueue"<<endl;
        cout<<"2 for Dequeue"<<endl;
        cout<<"3 for Peek"<<endl;
        cout<<"0 for Exit"<<endl;
        cin>>option;
        cout<<endl;

        if(option ==1){
            cout<<"Enter Value"<<endl;
            cin>>value;
            Q1.enqueue(value);
        }
        else if(option == 2){
            Q1.dequeue();
        }
        else if(option == 3){
            cout<<endl;
            Q1.peek();
        }
    }
}

```

}

```

C:\Users\SP22-BSCS-0046\Documents\Untitled6.exe
Enter Queue Size: 2
1 for Enqueue
2 for Dequeue
3 for Peek
0 for Exit
1
Enter Value
2
1 for Enqueue
2 for Dequeue
3 for Peek
0 for Exit
1
Enter Value
4
1 for Enqueue
2 for Dequeue
3 for Peek
0 for Exit
3
3
2
1 for Enqueue
2 for Dequeue
3 for Peek
0 for Exit
2
2
2
2
1 for Enqueue
2 for Dequeue
3 for Peek
0 for Exit
2
11867040

```

SOURCE CODE:

```
class Queue{
    private:
        int front;
        int rare;
        int arr_queue[];
        int queuesize;

    public:
        Queue(){
            front = -1;

```

```

        rare = -1;
    }
    Queue(int queuesize){
        arr_queue[queuesize];
    }
    void enqueueare(int value){
        if(front == -1){
            front = 0;
        }
        if(rare < queuesize-1){
            arr_queue[++rare] = value;
        }
    }
    void dequeuefront(){
        if(front != -1){
            cout<<arr_queue[front++];
        }
        else{
            cout<<"Queue is Empty : "<<endl;
        }
    }
    void peekfront(){
        if(front != -1){
            cout<<arr_queue[front];
        }
        cout<<"Queue is Empty : "<<endl;
    }
}

```

//

```

=====
=====

```

```

void enqueuefront(int value){
    if(rare == -1){
        rare = 0;
    }
    if(front < queuesize-1){
        arr_queue[++front] = value;
    }
}
void dequeuerare(){
    if(rare != -1 && rare < queuesize){

```

```

        cout<<arr_queue[rare--];
    }
    else{
        cout<<"Queue is Empty : "<<endl;
    }
}
void peekrare(){
    if(front != -1){
        cout<<arr_queue[front];
    }
    cout<<"Queue is Empty : "<<endl;
}
};

```

```

int main()
{
    Queue obj;

    int option;
    int value;

    cout<<"Enter the size of the Queue : "<<endl;
    cin>>value;

    do{
        cout<<"1. For Enqueuerare : "<<endl;
        cout<<"2. For DequeueFront : "<<endl;
        cout<<"3. For PeekFront : "<<endl;
        cout<<"4. For EnqueueFront : "<<endl;
        cout<<"5. For Dequeuerare : "<<endl;
        cout<<"6. For Peekrare : "<<endl;
        cout<<"7. For Exit : "<<endl;
        cin>>option;
        if(option == 1){
            cout<<"Enter the value : "<<endl;
            cin>>value;
            obj.enqueueare(value);
        }
        else if(option == 2){
            obj.dequeuefront();
        }
        else if(option == 3){
            obj.peekfront();
        }
    }
}

```

```

    }
    else if(option == 4){
        obj.enqueuefront(value);
    }
    else if(option == 5){
        obj.dequeueare();
    }
    else if(option == 6){
        obj.peekrare();
    }
    else{
        option = 0;
    }
}while(option != 0);

return 0;
}

```

PICTURE:

```

C:\Users\SP22-BSCS-0046\Documents\Untitled6.exe
5. For Enqueueare :
6. For Peekrare :
7. For Exit :
1
Enter the value :
2
1. For Enqueueare :
2. For DequeueFront :
3. For PeekFront :
4. For EnqueueFront :
5. For Dequeueare :
6. For Peekrare :
7. For Exit :
1
Enter the value :
4
1. For Enqueueare :
2. For DequeueFront :
3. For PeekFront :
4. For EnqueueFront :
5. For Dequeueare :
6. For Peekrare :
7. For Exit :
2
2.1. For Enqueueare :
2. For DequeueFront :
3. For PeekFront :
4. For EnqueueFront :
5. For Dequeueare :
6. For Peekrare :
7. For Exit :
2
4.1. For Enqueueare :
2. For DequeueFront :
3. For PeekFront :
4. For EnqueueFront :
5. For Dequeueare :
6. For Peekrare :
7. For Exit :
7
-----
Process exited after 53.1 seconds with return value 0
Press any key to continue . . .

```

4. Implement Stack using Queue.

SOURCE CODE:

PICTURE:

5. Implement Queue using Stack.

SOURCE CODE:

```

#include<iostream>
using namespace std;

```



```
const int n = 100;
```

```
class Stack {
```

```
private:
```

```
    int arr[n];
```

```
    int top;
```

```
public:
```

```
    Stack() {
```

```
        top = -1;
```

```
    }
```

```
    void push(int a){
```

```
        if(top<n-1){
```

```
            arr[++top] = a;
```

```
        }
```

```
        else{
```

```
            cout<<"Stack is Full";
```

```
        }
```

```
    }
```

```
    int peak(){
```

```
        return arr[top];
```

```
    }
```

```
    void pop(){
```

```
        if(top>-1){
```

```
            arr[top--];
```

```
        }
```

```
        else{
```

```
            cout<<"Stack is underflow"<<endl;
```

```
        }
```

```
    }
```

```
    bool empty(){
```

```
        if(top > -1){
```

```
            return 0;
```

```
        }
```

```
        else{
```

```
            return 1;
```

```
        }
```

```
    }
```

```
};
```

```
class queue{
```

```
    Stack s1,s2;
```

```
    public:
```

```
    void enqueue(int value){
```

```
        while(!s1.empty()){
```

```
            s2.push(s1.peak());
```

```
            s1.pop();
```

```
        }
```

```
        s1.push(value);
```

```
        while(!s2.empty()){
```

```
            s1.push(s2.peak());
```

```
            s2.pop();
```

```
        }
```

```
    }
```

```
    void dequeue(){
```

```
        if(s1.empty()){
```

```
            cout<< "Queue is Empty";
```

```
        }
```

```
        else{
```

```
            int x = s1.peak();
```

```
            s1.pop();
```

```
            cout<<x<<endl;
```

```
        }
```

```
    }
```

```
};
```

```
int main(){
```

```
    queue q;
```

```
    int option,a;
```

```
    while(true){
```

```
        cout<<"ENTER 1 FOR ENQUEUE"<<endl;
```

```
        cout<<"ENTER 2 FOR DEQUEUE"<<endl;
```

```

        cout<<"ENTER 0 FOR EXIT"<<endl;
        cout<<"ENTER YOUR CHOICE: "<<endl;
        cin>>option;
        if(option==1){
            cout<<"VALUE FOR ENQUEUE:"<<endl;
            cin>>a;
            q.enqueue(a);
        }
        else if(option==2){
            q.dequeue();
        }
        else if(option == 0){
            break;
        }
        else{
            cout<<"Wrong Input";
        }
    }

    return 0;
}

```

PICTURE:

```

C:\Users\SP22-BSCS-0046\Documents\Untitled1.exe
VALUE FOR ENQUEUE:
2
ENTER 1 FOR ENQUEUE
ENTER 2 FOR DEQUEUE
ENTER 0 FOR EXIT
ENTER YOUR CHOICE:
1
VALUE FOR ENQUEUE:
4
ENTER 1 FOR ENQUEUE
ENTER 2 FOR DEQUEUE
ENTER 0 FOR EXIT
ENTER YOUR CHOICE:
2
ENTER 1 FOR ENQUEUE
ENTER 2 FOR DEQUEUE
ENTER 0 FOR EXIT
ENTER YOUR CHOICE:
2
ENTER 1 FOR ENQUEUE
ENTER 2 FOR DEQUEUE
ENTER 0 FOR EXIT
ENTER YOUR CHOICE:
0
-----
Process exited after 48.87 seconds with return value 0
Press any key to continue . . .

```