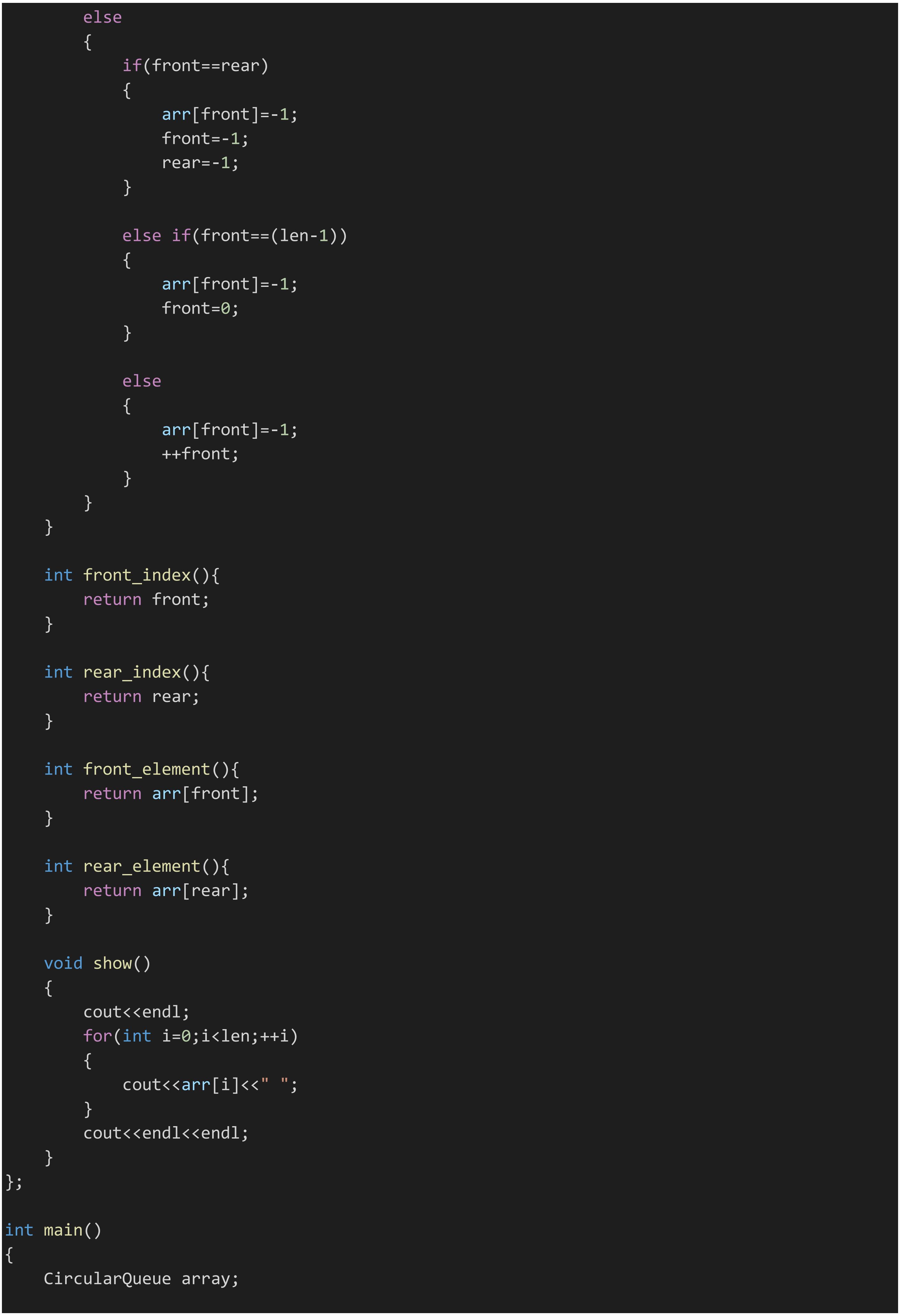
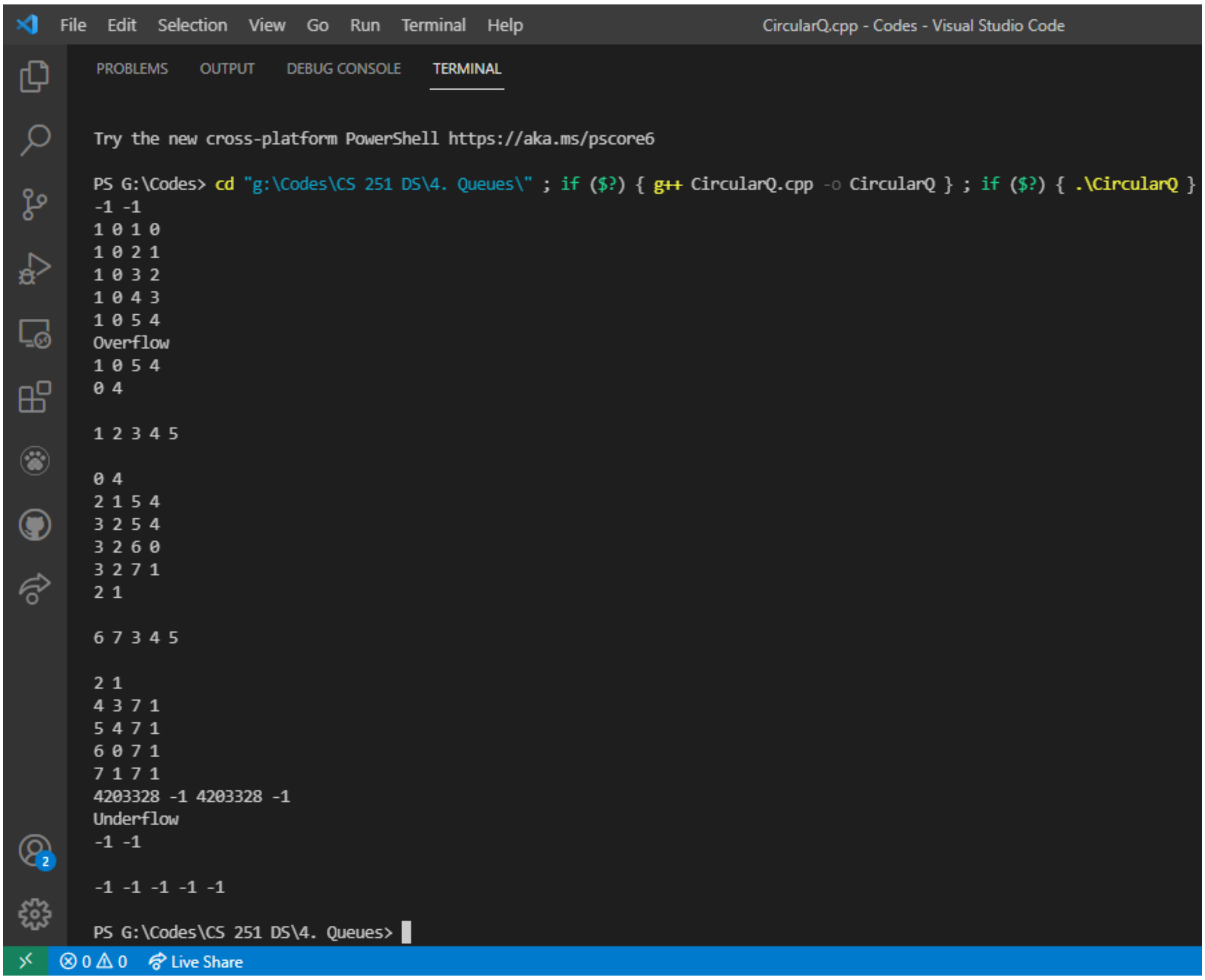
Circular Queue

```
// Aneesh Panchal
// 2K20/MC/21

#include<iostream>
using namespace std;

#define len 5
class CircularQueue
{
    public:
        int arr[len];
        int front=-1;
        int rear=-1;

    void enqueue(int element)
    {
        if(front==(rear+1) || front==(len-rear-1))
        {
            cout<<"Overflow"<<endl;
            return;
        }

        else
        {
            if(front==-1)
            {
                front=0;
                rear=0;
            }

            else if((rear==(len-1)) && (front!=0))
            {
                rear=0;
            }

            else
            {
                ++rear;
            }
            arr[rear]=element;
        }
    }

    void dequeue()
    {
        if(front==-1)
        {
            cout<<"Underflow"<<endl;
            return;
        }
```



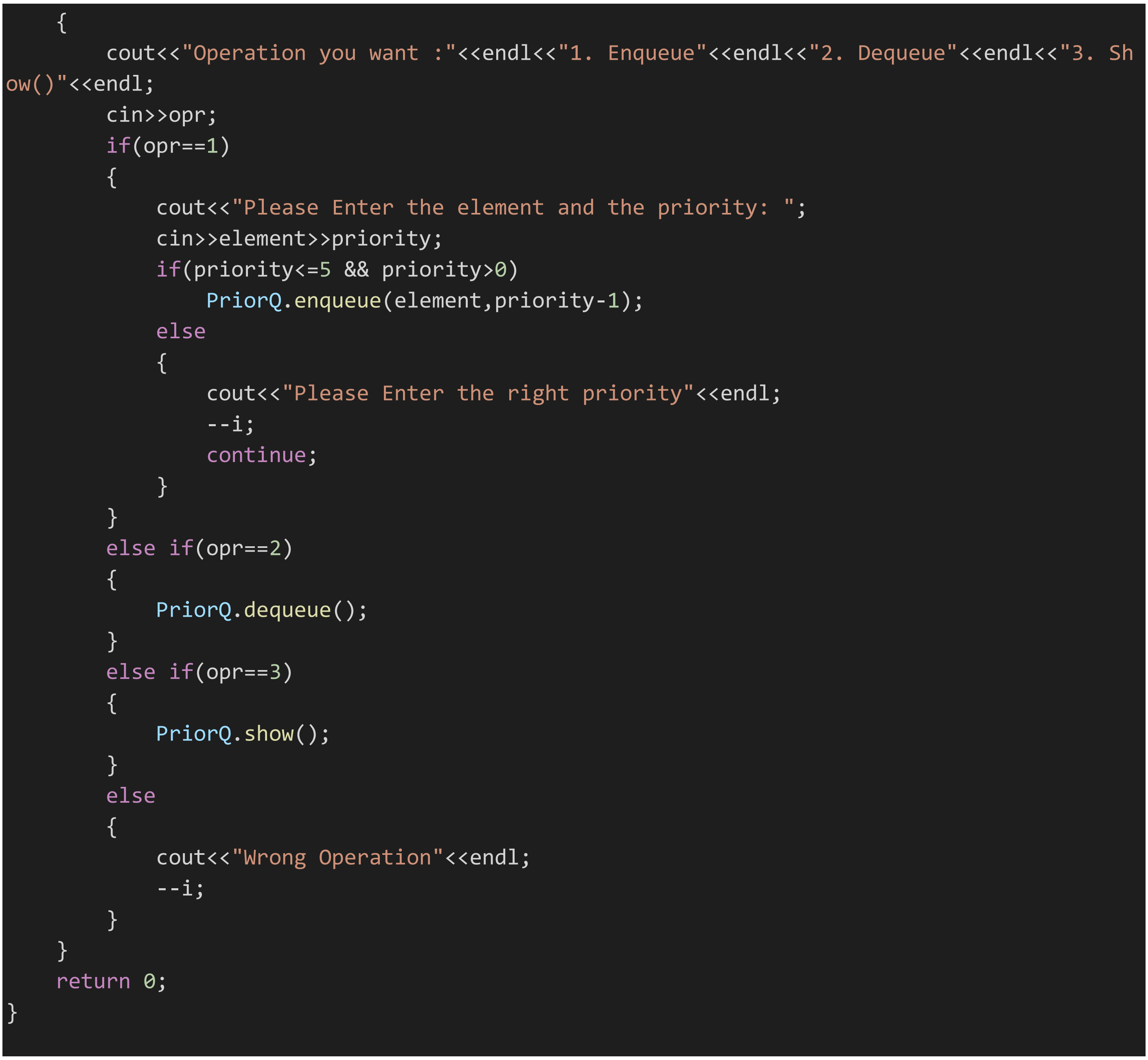
```cpp
        else
        {
            if(front==rear)
            {
                arr[front]=-1;
                front=-1;
                rear=-1;
            }

            else if(front==(len-1))
            {
                arr[front]=-1;
                front=0;
            }

            else
            {
                arr[front]=-1;
                ++front;
            }
        }
    }

    int front_index(){
        return front;
    }

    int rear_index(){
        return rear;
    }

    int front_element(){
        return arr[front];
    }

    int rear_element(){
        return arr[rear];
    }

    void show()
    {
        cout<<endl;
        for(int i=0;i<len;++i)
        {
            cout<<arr[i]<<" ";
        }
        cout<<endl<<endl;
    }
};

int main()
{
    CircularQueue array;
    cout<<array.front_index()<<" "<<array.rear_index()<<endl;

    array.enqueue(1);
    cout<<array.front_element()<<" "<<array.front_index()<<" "<<array.rear_element()<<" "<
<array.rear_index()<<endl;

    array.enqueue(2);
    cout<<array.front_element()<<" "<<array.front_index()<<" "<<array.rear_element()<<" "<
<array.rear_index()<<endl;

    array.enqueue(3);
    cout<<array.front_element()<<" "<<array.front_index()<<" "<<array.rear_element()<<" "<
<array.rear_index()<<endl;

    array.enqueue(4);
    cout<<array.front_element()<<" "<<array.front_index()<<" "<<array.rear_element()<<" "<
<array.rear_index()<<endl;

    array.enqueue(5);
    cout<<array.front_element()<<" "<<array.front_index()<<" "<<array.rear_element()<<" "<
<array.rear_index()<<endl;

    array.enqueue(6);
    cout<<array.front_element()<<" "<<array.front_index()<<" "<<array.rear_element()<<" "<
<array.rear_index()<<endl;
    cout<<array.front_index()<<" "<<array.rear_index()<<endl;

    array.show();

    cout<<array.front_index()<<" "<<array.rear_index()<<endl;

    array.dequeue();
    cout<<array.front_element()<<" "<<array.front_index()<<" "<<array.rear_element()<<" "<
<array.rear_index()<<endl;

    array.dequeue();
    cout<<array.front_element()<<" "<<array.front_index()<<" "<<array.rear_element()<<" "<
<array.rear_index()<<endl;

    array.enqueue(6);
    cout<<array.front_element()<<" "<<array.front_index()<<" "<<array.rear_element()<<" "<
<array.rear_index()<<endl;

    array.enqueue(7);
    cout<<array.front_element()<<" "<<array.front_index()<<" "<<array.rear_element()<<" "<
<array.rear_index()<<endl;
    cout<<array.front_index()<<" "<<array.rear_index()<<endl;

    array.show();

    cout<<array.front_index()<<" "<<array.rear_index()<<endl;

    array.dequeue();
    cout<<array.front_element()<<" "<<array.front_index()<<" "<<array.rear_element()<<" "<
<array.rear_index()<<endl;

    array.dequeue();
    cout<<array.front_element()<<" "<<array.front_index()<<" "<<array.rear_element()<<" "<
<array.rear_index()<<endl;

    array.dequeue();
    cout<<array.front_element()<<" "<<array.front_index()<<" "<<array.rear_element()<<" "<
<array.rear_index()<<endl;

    array.dequeue();
    cout<<array.front_element()<<" "<<array.front_index()<<" "<<array.rear_element()<<" "<
<array.rear_index()<<endl;

    array.dequeue();
    cout<<array.front_element()<<" "<<array.front_index()<<" "<<array.rear_element()<<" "<
<array.rear_index()<<endl;

    array.dequeue();
    cout<<array.front_index()<<" "<<array.rear_index()<<endl;

    array.show();

    return 0;
}
```


File Edit Selection View Go Run Terminal Help
CircularQ.cpp - Codes - Visual Studio Code
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
⊗ 0 ⚠ 0
Live Share

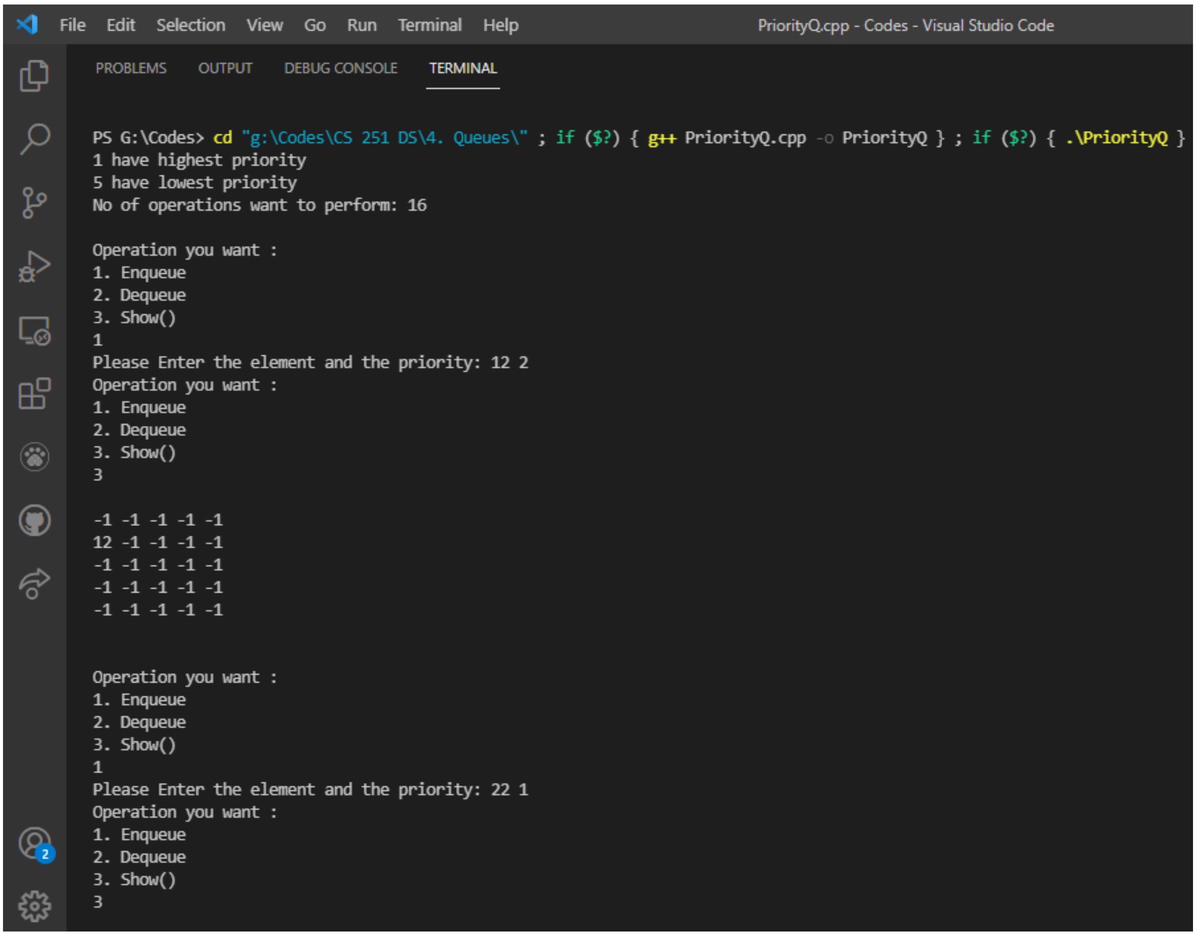
```
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS G:\Codes> cd "g:\Codes\CS 251 DS\4. Queues\" ; if ($?) { g++ CircularQ.cpp -o CircularQ } ; if ($?) { .\CircularQ }
-1 -1
1 0 1 0
1 0 2 1
1 0 3 2
1 0 4 3
1 0 5 4
Overflow
1 0 5 4
0 4

1 2 3 4 5

0 4
2 1 5 4
3 2 5 4
3 2 6 0
3 2 7 1
2 1

6 7 3 4 5

2 1
4 3 7 1
5 4 7 1
6 0 7 1
7 1 7 1
4203328 -1 4203328 -1
Underflow
-1 -1

-1 -1 -1 -1 -1

PS G:\Codes\CS 251 DS\4. Queues> 
```

Priority Queue

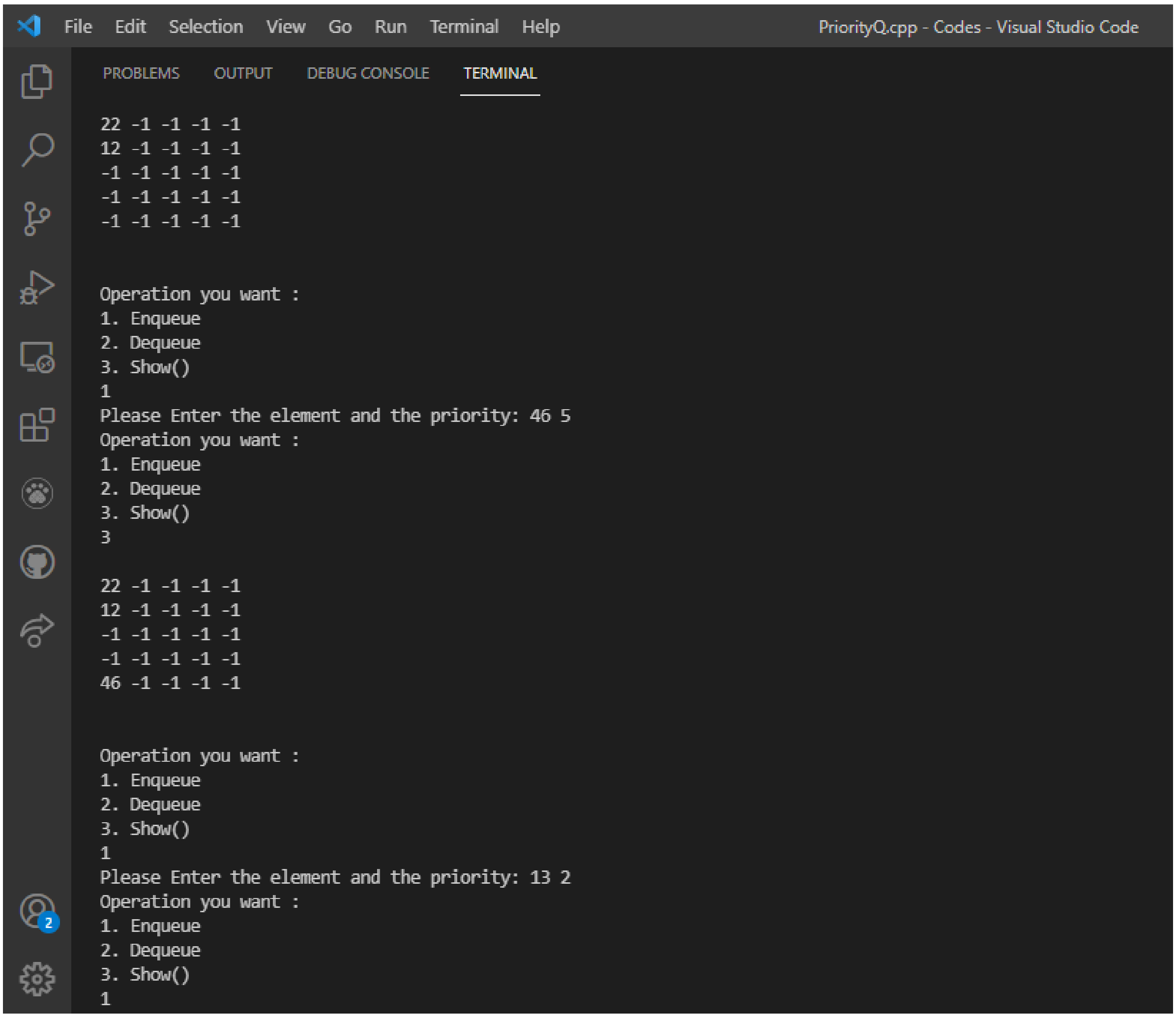
```cpp
// Aneesh Panchal
// 2K20/MC/21

#include<iostream>
using namespace std;

#define prior_len 5
#define len 5
class PriorityQueue
{
    public:
        int arr[prior_len][len]{-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-
1,-1,-1,-1,-1,-1};
        int front_rear[prior_len][2]={-1,-1,-1,-1,-1,-1,-1,-1,-1,-1};

    void enqueue(int element,int priority)
    {
        if(front_rear[priority][0]==(front_rear[priority][1]+1) || front_rear[priority][0]
==(len-front_rear[priority][1]-1))
        {
            cout<<"Overflow"<<endl;
            return;
        }

        else
        {
            if(front_rear[priority][0]==-1)
            {
                front_rear[priority][0]=0;
                front_rear[priority][1]=0;
            }

            else if((front_rear[priority][1]==(len-1)) && (front_rear[priority][0]!=0))
            {
                front_rear[priority][1]=0;
            }

            else
            {
                ++front_rear[priority][1];
            }
            arr[priority][front_rear[priority][1]]=element;
        }
    }

    void dequeue()
    {
        for(int i=0;i<prior_len;++i)
        {
            if(front_rear[i][0]!=-1)
            {
                if(front_rear[i][0]==front_rear[i][1])
                {
                    arr[i][front_rear[i][0]]=-1;
                    front_rear[i][0]=-1;
                    front_rear[i][1]=-1;
                }

                else if(front_rear[i][0]==(len-1))
                {
                    arr[i][front_rear[i][0]]=-1;
                    front_rear[i][0]=0;
                }

                else
                {
                    arr[i][front_rear[i][0]]=-1;
                    ++front_rear[i][0];
                }
                return;
            }
        }

        cout<<"Underflow"<<endl;
            return;
    }

    void show()
    {
        cout<<endl;
        for(int i=0;i<prior_len;++i)
        {
            for(int j=0;j<len;++j)
            {
                cout<<arr[i][j]<<" ";
            }
            cout<<endl;
        }
        cout<<endl<<endl;
    }

};

int main()
{
    cout<<"1 have highest priority"<<endl<<"5 have lowest priority"<<endl;

    PriorityQueue PriorQ;
    int element,priority,operations,opr;

    cout<<"No of operations want to perform: ";
    cin>>operations;
    cout<<endl;

    for(int i=0;i<operations;++i)
```



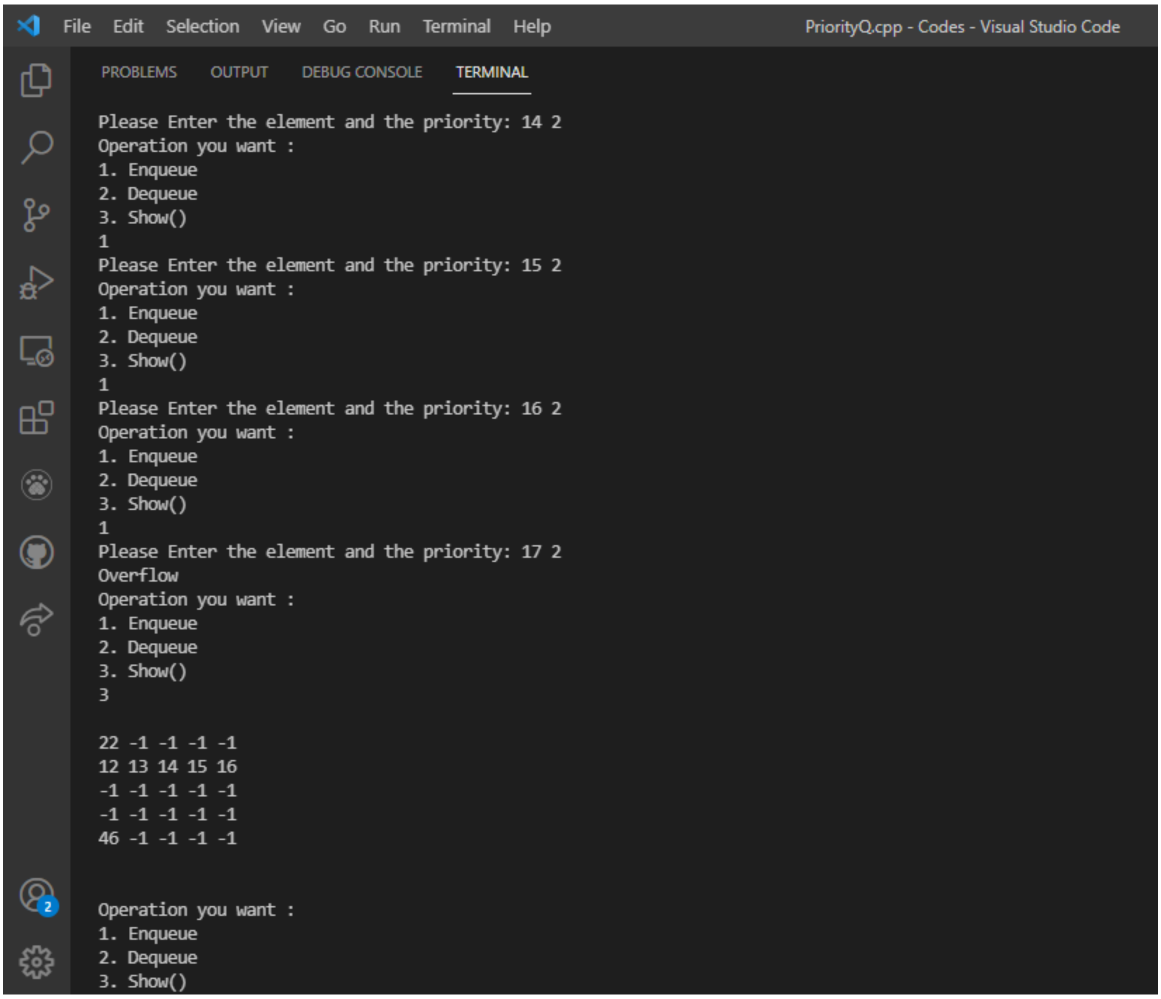
```
    {
        cout<<"Operation you want :"<<endl<<"1. Enqueue"<<endl<<"2. Dequeue"<<endl<<"3. Show()"<<endl;
        cin>>opr;
        if(opr==1)
        {
            cout<<"Please Enter the element and the priority: ";
            cin>>element>>priority;
            if(priority<=5 && priority>0)
                PriorQ.enqueue(element,priority-1);
            else
            {
                cout<<"Please Enter the right priority"<<endl;
                --i;
                continue;
            }
        }
        else if(opr==2)
        {
            PriorQ.dequeue();
        }
        else if(opr==3)
        {
            PriorQ.show();
        }
        else
        {
            cout<<"Wrong Operation"<<endl;
            --i;
        }
    }
    return 0;
}
```



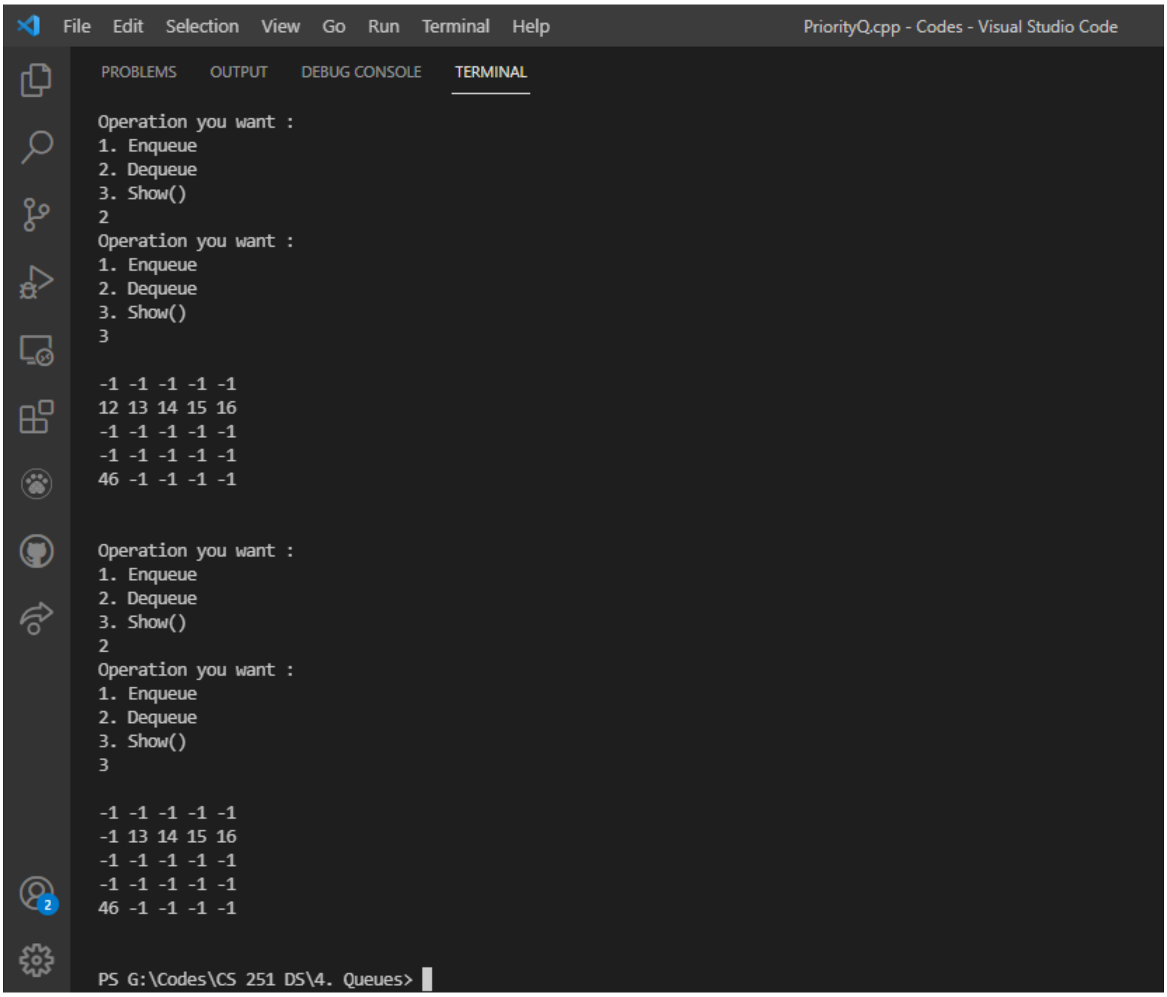
```
PS G:\Codes> cd "g:\Codes\CS 251 DS\4. Queues\" ; if ($?) { g++ PriorityQ.cpp -o PriorityQ } ; if ($?) { .\PriorityQ }
1 have highest priority
5 have lowest priority
No of operations want to perform: 16

Operation you want :
1. Enqueue
2. Dequeue
3. Show()
1
Please Enter the element and the priority: 12 2
Operation you want :
1. Enqueue
2. Dequeue
3. Show()
3

-1 -1 -1 -1 -1
12 -1 -1 -1 -1
-1 -1 -1 -1 -1
-1 -1 -1 -1 -1
-1 -1 -1 -1 -1


Operation you want :
1. Enqueue
2. Dequeue
3. Show()
1
Please Enter the element and the priority: 22 1
Operation you want :
1. Enqueue
2. Dequeue
3. Show()
3
```



```
22 -1 -1 -1 -1
12 -1 -1 -1 -1
-1 -1 -1 -1 -1
-1 -1 -1 -1 -1
-1 -1 -1 -1 -1


Operation you want :
1. Enqueue
2. Dequeue
3. Show()
1
Please Enter the element and the priority: 46 5
Operation you want :
1. Enqueue
2. Dequeue
3. Show()
3

22 -1 -1 -1 -1
12 -1 -1 -1 -1
-1 -1 -1 -1 -1
-1 -1 -1 -1 -1
46 -1 -1 -1 -1


Operation you want :
1. Enqueue
2. Dequeue
3. Show()
1
Please Enter the element and the priority: 13 2
Operation you want :
1. Enqueue
2. Dequeue
3. Show()
1
```


File Edit Selection View Go Run Terminal Help
PriorityQ.cpp - Codes - Visual Studio Code
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
Please Enter the element and the priority: 14 2
Operation you want :
1. Enqueue
2. Dequeue
3. Show()
1
Please Enter the element and the priority: 15 2
Operation you want :
1. Enqueue
2. Dequeue
3. Show()
1
Please Enter the element and the priority: 16 2
Operation you want :
1. Enqueue
2. Dequeue
3. Show()
1
Please Enter the element and the priority: 17 2
Overflow
Operation you want :
1. Enqueue
2. Dequeue
3. Show()
3

22 -1 -1 -1 -1
12 13 14 15 16
-1 -1 -1 -1 -1
-1 -1 -1 -1 -1
46 -1 -1 -1 -1


Operation you want :
1. Enqueue
2. Dequeue
3. Show()
```


File Edit Selection View Go Run Terminal Help
PriorityQ.cpp - Codes - Visual Studio Code
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
Operation you want :
1. Enqueue
2. Dequeue
3. Show()
2
Operation you want :
1. Enqueue
2. Dequeue
3. Show()
3

-1 -1 -1 -1 -1
12 13 14 15 16
-1 -1 -1 -1 -1
-1 -1 -1 -1 -1
46 -1 -1 -1 -1


Operation you want :
1. Enqueue
2. Dequeue
3. Show()
2
Operation you want :
1. Enqueue
2. Dequeue
3. Show()
3

-1 -1 -1 -1 -1
-1 13 14 15 16
-1 -1 -1 -1 -1
-1 -1 -1 -1 -1
46 -1 -1 -1 -1


PS G:\Codes\CS 251 DS\4. Queues>
```