FAIZAN CHOUDHARY

20BCS021

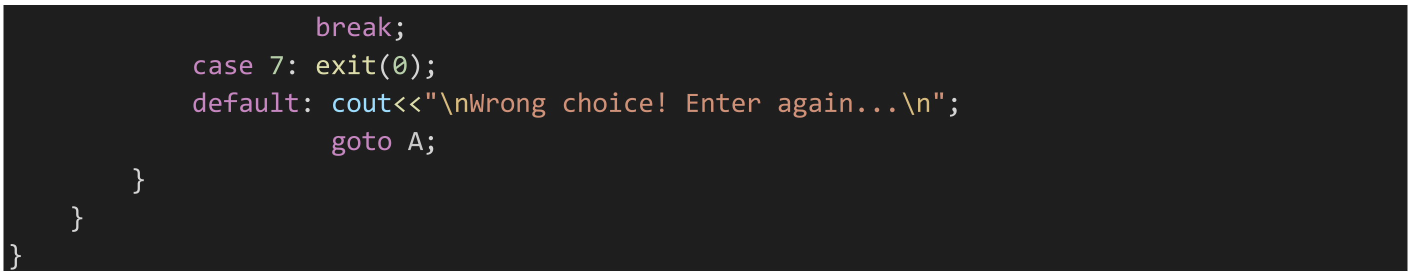
DSA LAB

30th November 2021

CODE: (code pasted in this format for readability)

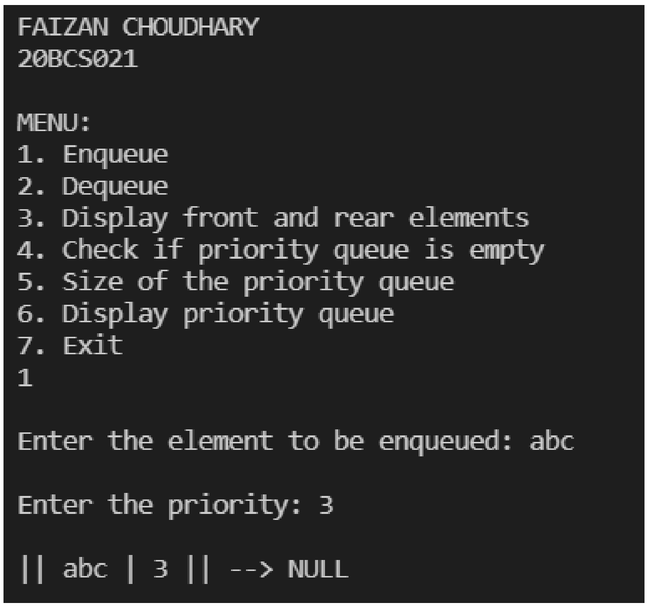
```
#include <iostream>
#include <string.h>
using namespace std;

struct PQueue
{
    char n[4];
    int pr;
    struct PQueue *next;
};
struct PQueue *front=NULL, *rear=NULL, *p, *ptr;

int isEmpty ()
{
    if (front==NULL)
        return 1;
    else
        return 0;
}

void display ()
{
    if (isEmpty()==1)
    {
        cout<<"\nPriority Queue is empty! Nothing to display\n";
        return;
    }

    else
    {
        p=front;
        cout<<endl;
        while (p->next!=NULL)
        {
            cout<<"|| "<<p->n<<" | "<<p->pr<<" || --> ";
            p=p->next;
        }
        cout<<"|| "<<p->n<<" | "<<p->pr<<" || --> NULL"<<endl;
    }
}

int size ()
```

```cpp
{
    int count=1;
    if (isEmpty()==1)
        return 0;
    else
    {
        p=front;
        while (p->next!=NULL)
        {
            count++;
            p=p->next;
        }
    }
    return count;
}

void front_rear ()
{

    if (isEmpty()==1)
    {
        cout<<"\nPriority Queue is empty..."<<endl;
        return;
    }
    p=front;
    while (p->next != NULL)
        p=p->next;
    rear=p;
    cout<<"\nFront element is: || "<<front->n<<" | "<<front->pr<<" ||";
    cout<<"\nRear element is: || "<<rear->n<<" | "<<rear->pr<<" ||"<<endl;
}

void enqueue (char* n, int pr)
{
    ptr=(struct PQueue *) malloc (sizeof(struct PQueue));
    if (ptr==NULL)
    {
        cout<<"\nMemory could not be allocated!\n";
        return;
    }
    strcpy(ptr->n, n);
    ptr->pr = pr;
    ptr->next=NULL;
    if (front==NULL || pr < (front->pr))
    {
        ptr->next = front;
        front=ptr;
    }
    else
    {
        p=front;
        while (p->next != NULL && p->next->pr <= pr)
            p=p->next;
        ptr->next = p->next;
        p->next = ptr;
    }

    display();
}

void dequeue ()
{
    if (isEmpty()==1)
        cout<<"\nPriority Queue Underflow!"<<endl;
    else
    {
        p = front;
        cout<<"\nDequeueing front element: || "<<p->n<<" | "<<p->pr<<" ||"<<endl;
        front=front->next;
        delete p;
        display();
    }
}

int main()
{
    cout<<"\nFAIZAN CHOUDHARY\n20BCS021\n";
    int ch,pr;
    char n[4];

    while (true)
    {
        A:
        cout<<"\nMENU:\n1. Enqueue\n2. Dequeue\n3. Display front and rear elements\n4.
Check if priority queue is empty\n5. Size of the priority queue\n6. Display priority
queue\n7. Exit\n";
        cin>>ch;
        switch (ch)
        {
            case 1: cout<<"\nEnter the element to be enqueued: ";
                    cin>>n;
                    cout<<"\nEnter the priority: ";
                    cin>>pr;
                    enqueue(n,pr);
                    break;
            case 2: dequeue();
                    break;
            case 3: front_rear();
                    break;
            case 4: if (isEmpty()==1)
                        cout<<"\nPriority Queue is empty!\n";
                    else
                        cout<<"\nPriority Queue is not empty.\n";
                    break;
            case 5: cout<<"\nSize of the priority queue is: "<<size()<<endl;
                    break;
            case 6: cout<<"\nPriority Queue elements: "<<endl;
                    display();
```

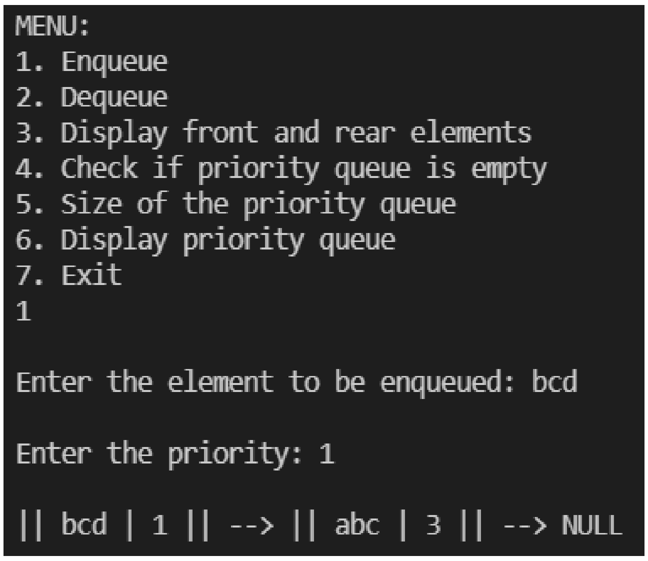


```
                    break;
            case 7: exit(0);
            default: cout<<"\nWrong choice! Enter again...\n";
                     goto A;
        }
    }
}
```

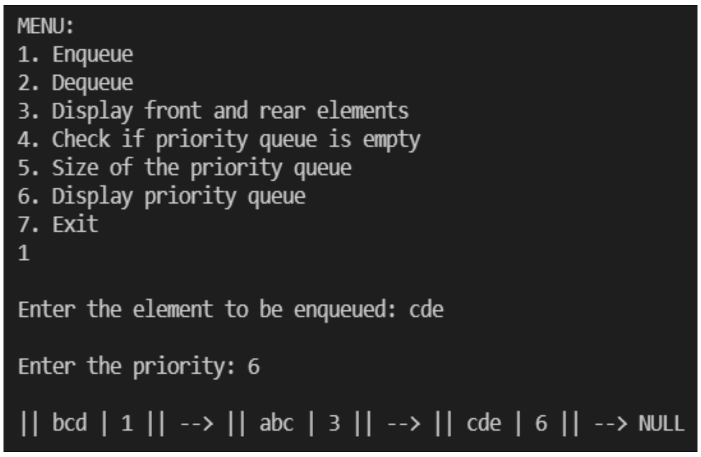
OUTPUT:



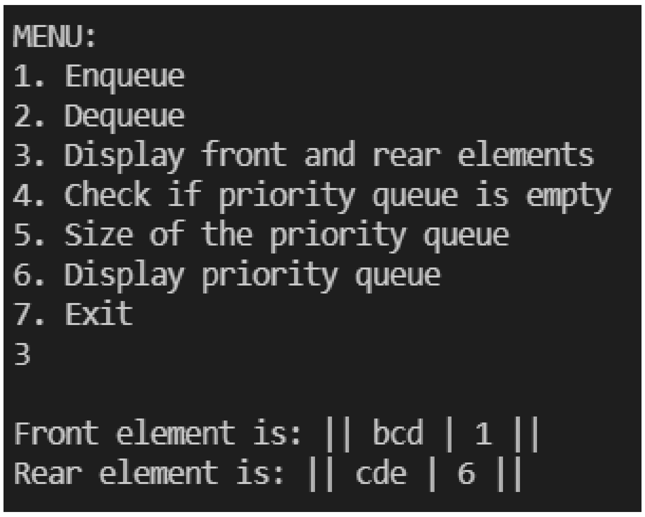
```
FAIZAN CHOUDHARY
20BCS021

MENU:
1. Enqueue
2. Dequeue
3. Display front and rear elements
4. Check if priority queue is empty
5. Size of the priority queue
6. Display priority queue
7. Exit
1

Enter the element to be enqueued: abc

Enter the priority: 3

|| abc | 3 || --> NULL
```



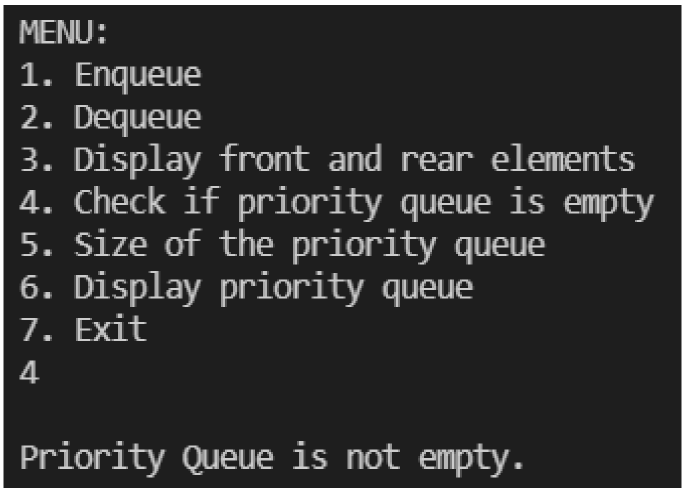
```
MENU:
1. Enqueue
2. Dequeue
3. Display front and rear elements
4. Check if priority queue is empty
5. Size of the priority queue
6. Display priority queue
7. Exit
1

Enter the element to be enqueued: bcd

Enter the priority: 1

|| bcd | 1 || --> || abc | 3 || --> NULL
```



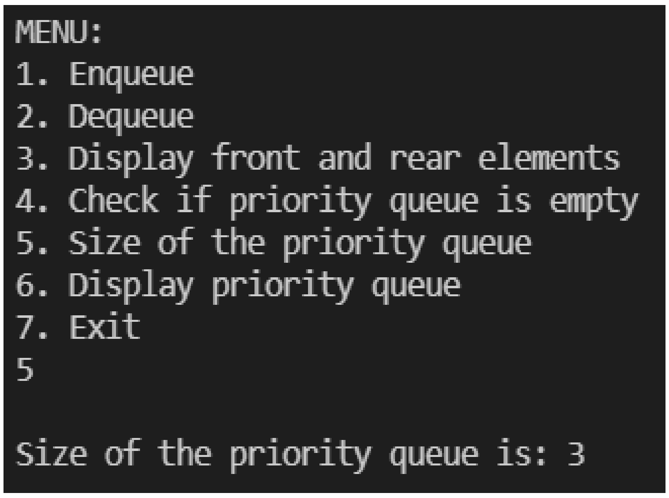
```
MENU:
1. Enqueue
2. Dequeue
3. Display front and rear elements
4. Check if priority queue is empty
5. Size of the priority queue
6. Display priority queue
7. Exit
1

Enter the element to be enqueued: cde

Enter the priority: 6

|| bcd | 1 || --> || abc | 3 || --> || cde | 6 || --> NULL
```



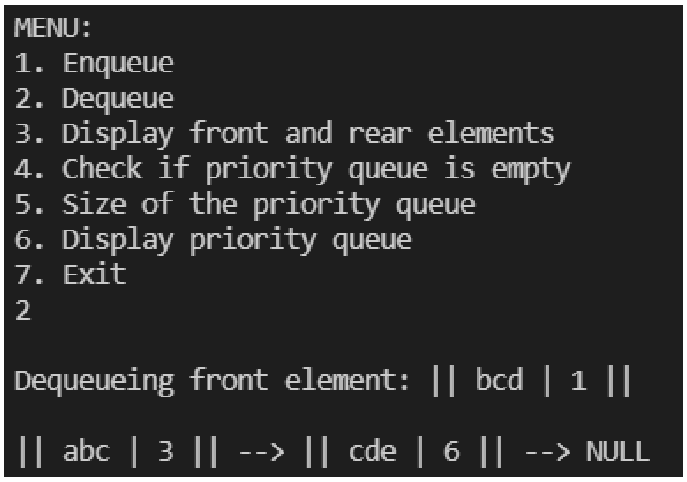
```
MENU:
1. Enqueue
2. Dequeue
3. Display front and rear elements
4. Check if priority queue is empty
5. Size of the priority queue
6. Display priority queue
7. Exit
3

Front element is: || bcd | 1 ||
Rear element is: || cde | 6 ||
```



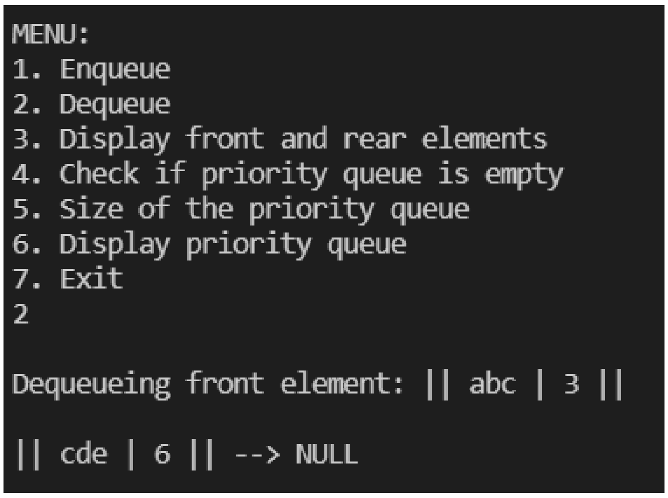
```
MENU:
1. Enqueue
2. Dequeue
3. Display front and rear elements
4. Check if priority queue is empty
5. Size of the priority queue
6. Display priority queue
7. Exit
4

Priority Queue is not empty.
```



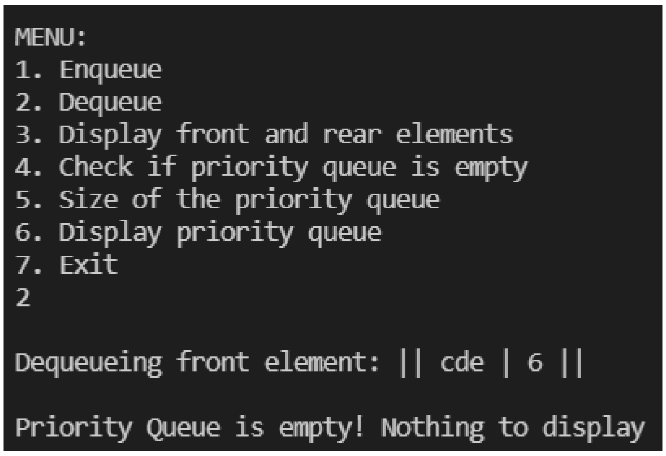
```
MENU:
1. Enqueue
2. Dequeue
3. Display front and rear elements
4. Check if priority queue is empty
5. Size of the priority queue
6. Display priority queue
7. Exit
5

Size of the priority queue is: 3
```



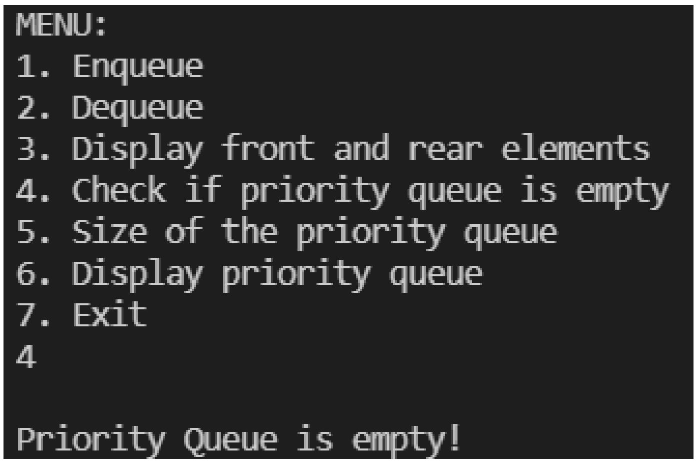
```
MENU:
1. Enqueue
2. Dequeue
3. Display front and rear elements
4. Check if priority queue is empty
5. Size of the priority queue
6. Display priority queue
7. Exit
2

Dequeueing front element: || bcd | 1 ||

|| abc | 3 || --> || cde | 6 || --> NULL
```



```
MENU:
1. Enqueue
2. Dequeue
3. Display front and rear elements
4. Check if priority queue is empty
5. Size of the priority queue
6. Display priority queue
7. Exit
2

Dequeueing front element: || abc | 3 ||

|| cde | 6 || --> NULL
```



```
MENU:
1. Enqueue
2. Dequeue
3. Display front and rear elements
4. Check if priority queue is empty
5. Size of the priority queue
6. Display priority queue
7. Exit
2

Dequeueing front element: || cde | 6 ||

Priority Queue is empty! Nothing to display
```



```
MENU:
1. Enqueue
2. Dequeue
3. Display front and rear elements
4. Check if priority queue is empty
5. Size of the priority queue
6. Display priority queue
7. Exit
4

Priority Queue is empty!
```