FAIZAN CHOUDHARY

20BCS021

OS LAB

20th January 2022

CODE: (code pasted in this format for readability)

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

struct PQueue
{
    char n[10];
    int pr;
    struct PQueue *next;
};
struct PQueue *front=NULL, *rear=NULL, *p, *ptr;

bool isEmpty () {
    if (front==NULL)
        return true;
    else
        return false;
}

void display () {
    if (isEmpty()==true) {
        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 totalProcess () {
    int count=1;
    if (isEmpty() == true)
        return 0;
    else {
        p=front;
```

```
        while (p->next!=NULL) {
            count++;
            p=p->next;
        }
    }
    return count;
}

void insertProcess (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 executeProcess () {
    if (isEmpty() == true)
        cout<<"\nPriority Queue Underflow!"<<endl;
    else {
        p = front;
        cout<<"\nExexcuted process is: || "<<p->n<<" | "<<p->pr<<" ||"<<endl;
        front=front->next;
        delete p;
        display();
    }
}

int main() {
    cout<<"\nFAIZAN CHOUDHARY\n20BCS021\n";
    int ch,pr;
    char n[10];

    while (true) {
        A:
        cout<<"\nMENU:\n1. Insert Process\n2. Execute Process\n3. Total number of
processes\n4. Display priority queue\n5. Exit\n";
        cin>>ch;
        switch (ch) {
```

```
            case 1: cout<<"\nEnter the process name: ";
                    cin>>n;
                    cout<<"\nEnter the priority: ";
                    cin>>pr;
                    insertProcess(n,pr);
                    break;
            case 2: executeProcess();
                    break;
            case 3: cout<<"\nTotal number of processes in priority queue are:
"<<totalProcess()<<endl;
                    break;
            case 4: cout<<"\nPriority Queue elements: "<<endl;
                    display();
                    break;
            case 5: exit(0);
            default: cout<<"\nWrong choice! Enter again...\n";
                     goto A;
        }
    }
    return 0;
}
```

OUTPUT:

```
FAIZAN CHOUDHARY
20BCS021

MENU:
1. Insert Process
2. Execute Process
3. Total number of processes
4. Display priority queue
5. Exit
1

Enter the process name: p4

Enter the priority: 5

|| p4 | 5 || --> NULL
```

```
MENU:
1. Insert Process
2. Execute Process
3. Total number of processes
4. Display priority queue
5. Exit
1

Enter the process name: p3

Enter the priority: 7

|| p9 | 2 || --> || p4 | 5 || --> || p3 | 7 || --> NULL
```

```
MENU:
1. Insert Process
2. Execute Process
3. Total number of processes
4. Display priority queue
5. Exit
1

Enter the process name: p9

Enter the priority: 2

|| p9 | 2 || --> || p4 | 5 || --> NULL
```

```
MENU:
1. Insert Process
2. Execute Process
3. Total number of processes
4. Display priority queue
5. Exit
1

Enter the process name: p6

Enter the priority: 5

|| p9 | 2 || --> || p4 | 5 || --> || p6 | 5 || --> || p3 | 7 || --> NULL
```

```
MENU:
1. Insert Process
2. Execute Process
3. Total number of processes
4. Display priority queue
5. Exit
3

Total number of processes in priority queue are: 4
```

```
MENU:
1. Insert Process
2. Execute Process
3. Total number of processes
4. Display priority queue
5. Exit
2

Exexcuted process is: || p9 | 2 ||

|| p4 | 5 || --> || p6 | 5 || --> || p3 | 7 || --> NULL
```

```
MENU:
1. Insert Process
2. Execute Process
3. Total number of processes
4. Display priority queue
5. Exit
2

Exexcuted process is: || p4 | 5 ||

|| p6 | 5 || --> || p3 | 7 || --> NULL
```

```
MENU:
1. Insert Process
2. Execute Process
3. Total number of processes
4. Display priority queue
5. Exit
5
```