Data Structure Lab

CEN-391

Program 10

Code :-

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

struct Priority_Queue
{
    char process[4];
    int priority;
    Priority_Queue *next;
};

void isEmpty(int size)
{
    cout << "isEmpty...\n";
    if (size == 0)
        cout << "Empty" << endl;
```

```
    else
        cout << "Not Empty" << endl;
}

void Display(Priority_Queue *head, int size)
{
    cout << "Display...\n";
    if (size == 0)
    {
        cout << "Queue Is Empty" << endl;
        return;
    }
    while (head != nullptr)
    {
        cout << head->process << "(" << head->priority <<
")"
             << "->";
        head = head->next;
    }
    cout << "Null\n";
    cout << endl;
}

void Enqueue(Priority_Queue *&head, Priority_Queue *&tail,
int &size)
{
    cout << "Enqueue...\n";
    Priority_Queue *newnode = (Priority_Queue *)malloc(1 *
sizeof(Priority_Queue));
    if (newnode == nullptr)
    {
        cout << "Memory Not Assigned" << endl;
        return;
    }
    size++;
    cout << "Enter The Priority : ";
    int priority;
    cin >> priority;
    fflush(stdin);
```

```
    cout << "Enter The Process Name : ";
    char process[4];
    gets(process);
    strcpy(newnode->process, process);
    newnode->priority = priority;
    newnode->next = nullptr;

    Priority_Queue *temp = head;
    if (head == nullptr)
    {
        head = newnode;
        tail = newnode;
    }
    else
    {
        if (temp->priority > newnode->priority)
        {
            newnode->next = head;
            head = newnode;
        }
        else if (tail->priority <= newnode->priority)
        {
            tail->next = newnode;
            tail = tail->next;
        }
        else
        {
            while (temp && temp->next)
            {
                if (temp->next->priority > newnode-
>priority)
                {
                    newnode->next = temp->next;
                    temp->next = newnode;
                    break;
                }
                temp = temp->next;
            }
        }
```

```
    }
    Display(head, size);
}

void Dequeue(Priority_Queue *&head, int &size)
{
    cout << "Dequeue...\n";
    if (size == 0)
    {
        cout << "Queue Underflow" << endl;
        return;
    }
    cout << head->process << "(" << head->priority << ")"
         << "\n";
    size--;
    Priority_Queue *todelete = head;
    head = head->next;
    delete todelete;
    Display(head, size);
}
void Front_Rear(Priority_Queue *head, Priority_Queue *tail,
int size)
{
    cout << "Front And Rear...\n";
    if (size == 0)
    {
        cout << "Queue Is Empty" << endl;
        return;
    }
    cout << "Front : " << head->process << endl;
    cout << "Rear : " << tail->process << endl;
}

void Total_Element(int size)
{
    cout << "Total Elements In Priority Queue : " << size <<
endl;
}
```

```
void Bars()
{
    cout << "-----------------------------------------------
----------------\n";
}
bool Options(Priority_Queue *&head, Priority_Queue *&tail,
int &size)
{
    int opt;
    cin >> opt;
    Bars();
    switch (opt)
    {
    case 1:
        Enqueue(head, tail, size);
        break;
    case 2:
        Dequeue(head, size);
        break;
    case 3:
        Front_Rear(head, tail, size);
        break;
    case 4:
        isEmpty(size);
        break;
    case 5:
        Total_Element(size);
        break;
    case 6:
        Display(head, size);
        break;
    case 7:
        cout << "Exit...\n";
        return 0;
    default:
        cout << "Invalid Input!\nTry Again!\n";
    }
    Bars();
    return 1;
```

```
}

void Menu()
{
    cout << "_____Operations_On_Priority_Queue_____ \n";
    cout << "1.Enqueue \n";
    cout << "2.Dequeue \n";
    cout << "3.Front And Rear Element \n";
    cout << "4.isEmpty \n";
    cout << "5.Total Elements \n";
    cout << "6:Display \n";
    cout << "7.Exit \n";
    cout << "Enter Your Choice : ";
}

int main()
{

    system("cls");
    cout << "_____Vicky_Gupta_20BCS070_____\n\n";
    int size = 0;
    Priority_Queue *head = nullptr, *tail = nullptr;
    while (true)
    {
        Menu();
        if (!Options(head, tail, size))
            break;
    }
    cout << "Exiting...\n";
    Bars();
    return 0;
}
```

Output :-

```
_____Vicky_Gupta_20BCS070_____

_____Operations_On_Priority_Queue_____
1.Enqueue
2.Dequeue
3.Front And Rear Element
4.isEmpty
5.Total Elements
6:Display
7.Exit
Enter Your Choice : 1
-----------------------------------------
Enqueue...
Enter The Priority : 5
Enter The Process Name : P1
Display...
P1(5)->Null

-----------------------------------------
_____Operations_On_Priority_Queue_____
1.Enqueue
2.Dequeue
3.Front And Rear Element
4.isEmpty
5.Total Elements
6:Display
7.Exit
Enter Your Choice : 1
-----------------------------------------
Enqueue...
Enter The Priority : 4
Enter The Process Name : P2
Display...
P2(4)->P1(5)->Null

-----------------------------------------
```

```
_____Operations_On_Priority_Queue_____
1.Enqueue
2.Dequeue
3.Front And Rear Element
4.isEmpty
5.Total Elements
6:Display
7.Exit
Enter Your Choice : 1
----------------------------------------
Enqueue...
Enter The Priority : 6
Enter The Process Name : P3
Display...
P2(4)->P1(5)->P3(6)->Null

----------------------------------------
_____Operations_On_Priority_Queue_____
1.Enqueue
2.Dequeue
3.Front And Rear Element
4.isEmpty
5.Total Elements
6:Display
7.Exit
Enter Your Choice : 3
----------------------------------------
Front And Rear...
Front : P2
Rear : P3
----------------------------------------
```

```
_____Operations_On_Priority_Queue_____
1.Enqueue
2.Dequeue
3.Front And Rear Element
4.isEmpty
5.Total Elements
6:Display
7.Exit
Enter Your Choice : 2
-------------------------------------------------
Dequeue...
P2(4)
Display...
P1(5)->P3(6)->Null

-------------------------------------------------
_____Operations_On_Priority_Queue_____
1.Enqueue
2.Dequeue
3.Front And Rear Element
4.isEmpty
5.Total Elements
6:Display
7.Exit
Enter Your Choice : 5
-------------------------------------------------
Total Elements In Priority Queue : 2
-------------------------------------------------
_____Operations_On_Priority_Queue_____
1.Enqueue
2.Dequeue
3.Front And Rear Element
4.isEmpty
5.Total Elements
6:Display
7.Exit
Enter Your Choice : 2
-------------------------------------------------
Dequeue...
P1(5)
Display...
P3(6)->Null
```

```
_____Operations_On_Priority_Queue_____
1.Enqueue
2.Dequeue
3.Front And Rear Element
4.isEmpty
5.Total Elements
6:Display
7.Exit
Enter Your Choice : 6
-----------------------------------------
Display...
P3(6)->Null

-----------------------------------------
_____Operations_On_Priority_Queue_____
1.Enqueue
2.Dequeue
3.Front And Rear Element
4.isEmpty
5.Total Elements
6:Display
7.Exit
Enter Your Choice : 2
-----------------------------------------
Dequeue...
P3(6)
Display...
Queue Is Empty
-----------------------------------------
_____Operations_On_Priority_Queue_____
1.Enqueue
2.Dequeue
3.Front And Rear Element
4.isEmpty
5.Total Elements
6:Display
7.Exit
Enter Your Choice : 4
-----------------------------------------
isEmpty...
Empty
-----------------------------------------
```

```
_____Operations_On_Priority_Queue_____
1.Enqueue
2.Dequeue
3.Front And Rear Element
4.isEmpty
5.Total Elements
6:Display
7.Exit
Enter Your Choice : 7
----------------------------------------
Exit...
Exiting...
----------------------------------------
```