ASSIGNMENT-01

Name: K V Jaya Harsha

Roll no: CS23B1034 Date: 22-08-2024

Q1. Priority Queue. (in cpp)

```cpp
//CS23B1034
//K V Jaya Harsha
#include <iostream>
using namespace std;
class PriorityQueue
{
private:
    int size, capacity, *elements, *priorities;
public:
    PriorityQueue(int cap) : size(0), capacity(cap){
        elements = new int[capacity];
        priorities = new int[capacity];
    }
    ~PriorityQueue(){
        delete[] elements;
        delete[] priorities;
    }
    void enqueue(int element, int priority){
        if (size == capacity){
            cout << "Queue Overflow!" << endl;
            return;
        }
        elements[size] = element;
        priorities[size] = priority;
        size++;
        insertionSort();
    }
    int dequeue(){
        if (size == 0){
            cout << "Queue Underflow!" << endl;
            return -1;
        }
        int dequeuedElement = elements[0];
        for (int i = 1; i < size; i++){
            elements[i - 1] = elements[i];
            priorities[i - 1] = priorities[i];
        }
        size--;
        return dequeuedElement;
    }

    int front() const{
        if (size == 0){
            cout << "Queue Underflow!" << endl;
            return -1;
        }
        return elements[0];
    }
    void display() const{
        if (size == 0){
            cout << "Queue Underflow!" << endl;
            return;
        }
        cout << "Elements in queue: " << endl;
        for (int i = 0; i < size; i++){
            cout << "Element: " << elements[i] << " => Priority: " << priorities[i] << endl;
        }
    }
```

```
private:
    void insertionSort(){
        for (int i = 1; i < size; i++){
            int keyElement = elements[i];
            int keyPriority = priorities[i];
            int j = i - 1;

            while (j >= 0 && priorities[j] > keyPriority){
                elements[j + 1] = elements[j];
                priorities[j + 1] = priorities[j];
                j--;
            }
            elements[j + 1] = keyElement;
            priorities[j + 1] = keyPriority;
        }
    }
};
int main(){
    PriorityQueue pq(5);
    pq.enqueue(10, 2);
    pq.enqueue(20, 1);
    pq.enqueue(30, 3);
    pq.enqueue(40, 0);
    pq.display();
    cout << "Element at front: " << pq.front() << endl;
    pq.dequeue();
    pq.dequeue();
    pq.display();
    return 0;
}
```

```
 klnvdx } ; if ($?) { .\klnvdx
Elements in queue:
Element: 40 => Priority: 0
Element: 20 => Priority: 1
Element: 10 => Priority: 2
Element: 30 => Priority: 3
Element at front: 40
Elements in queue:
Element: 10 => Priority: 2
Element: 30 => Priority: 3
PS C:\Users\harsh\OneDrive\Docu
```