CA1 of CAP 770 Advance Data Structures

Name: Narra Suryakoushik Reddy
IDno: 12403164
Roll no: 24
Section: D2422(MCA)

Question given by the teacher to implement the queues for the given question without STL.

Answer:

```
#include<iostream>
using namespace std;

class Queue{
   struct Node{
       string data;
       Node* next;
       Node(string val) : data(val), next(nullptr){}
   };
   Node*front;
   Node*back;
public:
   Queue() : front(nullptr), back(nullptr){}

   void enqueue(string value){
       Node* newNode = new Node(value);
       if(back == nullptr){
           front = back = newNode;
       }
       else{
           back-> next = newNode;
           back = newNode;
       }
   }
   void teleport(){
       if(front == nullptr){
           cout<<"queue is empty\n";
           return;
       }
       Node* temp = front;
       front = front->next;
       if(front ==nullptr){
           back=nullptr;
```

```
        }
        delete temp;
    }
    string peek(){
        if(front == nullptr){
            return "queue is empty\n";
        }
        return front->data;
    }
    bool is_empty(){
        return front==nullptr;
    }
    void display(){
        Node* temp = front;
        while(temp){
            cout<<temp->data<<",  ";
            temp= temp->next;
        }
        cout<<endl;
    }

    ~Queue(){
        while (!is_empty()){
            teleport();
        }
    }


};
int main() {
    Queue q;
    q.enqueue("elf");
    q.enqueue("dwarf");
    q.enqueue("fairy");
    q.enqueue("goblin");
    q.enqueue("centaur");

    cout<<"before teleporting: ";
    q.display();

    Queue tempQueue;
    string requeue1 = "fairy", requeue2 = "goblin";
    Queue finalQueue;
```

```
    while (!q.is_empty()) {
        string creature = q.peek();
        q.teleport();

        if (creature == requeue1 || creature == requeue2) {
            tempQueue.enqueue(creature);
        } else {
            finalQueue.enqueue(creature);
        }
    }

    while (!tempQueue.is_empty()) {
        finalQueue.enqueue(tempQueue.peek());
        tempQueue.teleport();
    }

    cout << "queue after re-enqueuing misrouted creatures: ";
    finalQueue.display();

    return 0;


}
```

Output:

```
❯ cd "/home/reddy/codes/DSA/CA1Speedrun/" && g++ DSAca1.cpp -o DSAca1 && "/home/reddy/codes/DSA/CA1Speedrun/"DSAca1
before teleporting: elf,  dwarf,  fairy,  goblin,  centaur,
queue after re-enqueuing misrouted creatures: elf,  dwarf,  centaur,  fairy,  goblin,
```

The output is as required and the code was successfully executed by giving the expected output of requeuing the misteleported creatures.