Lab 04 [CO3]

Greetings Students. In this lab, we will play with Dummy Headed Doubly Circular Linked List. If you want to read about this type of linked list then check **this file.**

In this lab, you have to implement a waiting room management system in an emergency ward of a hospital. Your program will serve a patient on a **first-come-first-serve basis.**

Solve the above problem using a **Dummy Headed Doubly Circular Linked List.**

1. You need to have a **Patient** class so that you can create an instance of it (patient) by assigning id(integer), name (String), age (integer), and blood group (String).

2. Write a **WRM** (waiting room management) class that will contain the below methods.
    a. **RegisterPatient(id, name, age, bloodgroup):** This method will register a patient into your system. The method will create a Patient type object with the information received as parameter. It means this method will add a patient-type object to your linked list.
    b. **ServePatient():** This method calls a patient to provide hospital service to him/her. In this method, you need to ensure to serve the patient first who was registered first.
    c. **CancelAll():** This method cancels all appointments of the patients so that the doctor can go to lunch.
    d. **CanDoctorGoHome():** This method returns true if no one is waiting, otherwise, returns false.
    e. **ShowAllPatient():** This method prints all names of the waiting patients in sequential order. It means the patient who got registered first, will come first, and so on.
    f. **ReverseTheLine():** This method reverses the patient line. It means the patient who got registered last, will come first, and so on.

3. Write a **Tester** code that will interact with users and take information about Patients. You will pass this information to **WRM** and create instances of **Patient** in **WRM** and call the methods of **WRM** class. You just need to ensure your Tester code has completed all the properties mentioned in 4 no point.

4. Tester Code Options:
    a. Add Patient – print Success or Not
    b. Serve Patient – print Name of Patient being Served
    c. Show All patients – print all patient in sequence to serve
    d. Can Doctor go Home? – return yes or no
    e. Cancel all Appointment – print Success or Not
    f. ReverseTheLine - print Success or Not

Hints:

Usual Node class design in doubly linked list:

```
class DoublyNode:
        def __init__(self, elem, next, prev):
                self.elem = elem
                self.next = next # To store the next node's reference.
                self.prev = prev # To store the previous node's reference.
```

In your program your Patient class will work as the Node class for the Dummy Headed Doubly Circular Linked List and WRM class will work as that Linked List.