INHA University in Tashkent

Data Structure, fall 2017

Assignment 1

Submission deadline: 23:59, October 5, 2017

Write C++ code for the given task and upload your solution (CPP file) to E-Class.

INHA University in Tashkent want to store it's employees record (ID, Name, Date of Birth, Phone number and address) and perform several operation like searching, adding, deleting, modifying, display info etc.

You are require to write C++ Implementation of linked list (singly) data structures for this problem. When your code is executed it should give the following options to the user and perform the actions according to the user selection

1. Display Front Employee's Info
2. Display Back Employee's Info
3. Print Employee's List (all info)
4. Insert an Employee at Front
5. Insert an Employee at Back
6. Delete Employee at the Front
7. Delete Employee at the Back
8. Search for an Employee (based on Employee's ID)
9. Modify an Employee data (Name, Date of Birth, Phone, address)
10. Add a New Employee (order of ID)
11. Delete an Employee
12. Count Employees

Your implementation should contain the following two classes, at least.

- Class **Employee** for instantiating nodes.
- Class **EmpLinkedList** that contain the head pointer and the following member functions (Use appropriate return type and arguments for each function).

- **IsEmpty ()**
  - Check if the list is empty
- **Front()**
  - Return the front node
- **Back()**
  - Return the Back node
- **Print_List()**
  - Display all elements (Complete info of all employees)

- **AddFront()**
  - Insert Node at the front of the list

- **AddBack()**
  - Insert Node at the back
- **DeleteFront()**
  - Delete the front Node
- **DeleteBack()**
  - Delete the last Node
- **Find()**
  - Find the Node (with given ID) in the list and display its data
- **Modify()**

  Modify the content (Name, Date of Birth, Phone, address)

  - of a given node
- **Add()**
  - Insert a new node at the appropriate location (based on Employee ID, it could be in the middle or even front or back of the list)
- **Delete()**
  - Delete a given node, It could be located anywhere in the list (front, back middle)
- **Count()**
  - Count the number of nodes in the list