ATILIM UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING

CMPE 226 Data Structures
2021 - 2022 Summer, Homework II
Instructors: Zühal Kurt, Murat Karakaya
Due Date: 07.07.2022 / 23:59

RULES & REGULATIONS

- You must **submit** your homework **via Moodle before the due date/time**.
- Homework submitted via email will be **ignored**!
- **Late submissions will NOT be graded.**
- **Before submission check that**:
  - You can use any IDE you want, but we prefer to you use **CodeBlocks** or **Embarcadero Dev-C++**
  - Set your language standard to **ISO C++11**
  - Name your homework file as in the format **lastname_firstname_hw2.cpp**.
  - **Do NOT use Turkish characters when you name C++ source file.**
  - **Do NOT upload .exe file. Otherwise you will get ZERO.**
- **Your codes** will be checked by special software (JPLAG & MOSS) for code similarity. If the **code similarity** between any two or more submissions is **higher than %90**, we will also examine and compare these codes by eye. If we are convinced that the similarity between two codes is not merely a coincidence, **all involved HWs will get 0 as the grade**.
- **Therefore**;
  - **Group study is not allowed.** Everyone needs to do his/her homework as an individual.
  - **Be careful when you use Internet resources! Not use them directly!** If any other student uses the same source then your submissions will be found as similar!
  - **Do not share your solution/ideas with others.**
- Moreover, if we believe that the students cheat on these exams, we will initiate a **disciplinary act**!
- **GRADING POLICY**: If your code has a compiler or runtime error caused by major mistakes, your code will be evaluated over **50 points not 100 points**.

Homework Definition

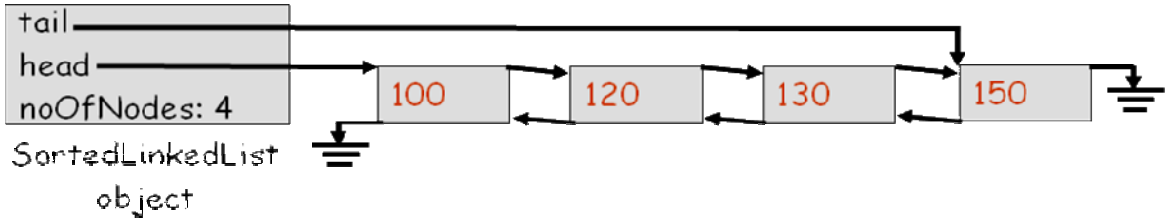
In this project you are to implement a **SortedLinkedList** class which maintains the list elements in ascending order of key values in a **doubly** linked list.
Each list node contains a key and a pointer to the previous and next nodes in the list as follows:

```
/* LinkedListNode */
public class LinkedListNode {
  public int key;          /* Key */
  public LinkedListNode prev;   /* Pointer to the
  previous node */public LinkedListNode next;   /*
  Pointer to the next node */
};
```

SortedLinkedList class has 3 variables and is defined as follows:

```
public class SortedLinkedList {
  public LinkedListNode head, tail;/* Head and tail of the
  list of nodes */public int noOfNodes;   /* Current number of
  nodes in the list */
};
```

head points to the beginning of the doubly linked list, **tail** points to the end of the doubly linked list and **noOfNodes** counts the current nodes in the list. The figure below shows a snapshot of an example doubly linked list that contains the keys 100, 120, 130 and 150. Clearly the keys are maintained in ascending order oftheir values.



1 How to Implement the SortedArrayList Class

Public interface of SortedLinkedList class specified in **SortedLinkedList.h** are given below

```
  /*******************************************************
   * Constructor: Initializes the linked list
   ********************************************/
  SortedLinkedList();

  /*******************************************************
   * Removes all of the nodes from this list.
   ********************************************/
  void Clear();

  /*******************************************************
   * Returns the number of nodes in the list
   ********************************************/
  int NoOfNodes(){return noOfNodes;}


  /*********************************************
*     Inserts the given key in ascending order in
    the list
   *********************************************/
  public void Add(int key);

  /*********************************************
*     Removes the node that contains the key from
    the list
   * (if the key is found)
   * Returns 0 upon successful deletion, -1 on error
   *********************************************/
  public void Remove(int key);
```

```
/*****************************************************
 * Removes a node from the list given a pointer to the node
 *****************************************/
public void Remove(LinkedListNode node);

/*****************************************************
 * Searches a key in the list and returns a pointer
 * to the list node that contains the key
 *****************************************/
public LinkedListNode Find(int key);
```

In main function, create a SortedLinkedList object and show the usage of each public member functions. Write all the definitions of each public member functions which are given above. Finally get this output after the sample run.

Sample Run:

Created SortedLinkedList is:
Print all values in ascending order
20 30 60 70 90
Print all values in descending order
90 70 60 30 20
Forward Index of 30 is: 2
Backward Index of 30 is: 4
SortedLinkedList includes 5 number of nodes