EADS LAB TASK1

Public Methods:

Sequence()	**Constructor: Creates an Empty List**
~Sequence()	**Destructor: Deletes whole list.**
Sequence(const Sequence<Key,Info> &x)	**Copy Constructor: Copies the Sequence to the new defined done.**
Sequence<Key,Info> &operator=(const Sequence<Key,Info> &x)	**Assignment Operator: Copies one Sequence to predefined the other.**
bool operator==(const Sequence<Key,Info> &x)	**Comparison Operator: returns true if both sequences are equal,false if not.**
bool operator!=(const Sequence<Key,Info> &x)	**Comparison Operator: returns false if both sequences are equal,true if not.**
bool pushBack(Info data,Key key)	**Pushes the given element as the last element of the sequence.**
bool pushFront(Info data,Key key)	**Pushes the given element as the first element of the sequence.**
node findElement(int pos) const	**Finds the element in the given position and returns its pointer. Returns NULL if there is no element in that position.**
bool removeByKey(Key key)	**Removes all elements with the given key. Returns true if atleast 1 element is deleted,false if not.**
bool removeByInfo(Info data)	**Removes all elements with the given Info. Returns true if atleast 1 element is deleted,false if not.**
bool removeByPos(int pos)	**Removes the element with the given position in the list**
int NumberOfElements() const	**Returns number of the elements in the list.**
bool isEmpty()	**Returns true if first node is NULL(list is empty) ,false if not.**
void clear()	**Deletes all the elements from the list**
void print()	**Prints the whole list with spaces between them.**

Private:

```
typedef struct Node          This way I can store both data and key
   {                         in the sequence.
      Info data;
      Key key;
      Node *next;         → Pointer of the next element.
   }*node;
```

```
node first;    → Pointer of the first element(NULL at first)
node last;     → Pointer of the last element(NULL at first)
int numberOfElements;   → Number of elements in the list(0 at first)
```

MERT ŞAMİL GÜL   K-5379