

# LABORATORIES 1 EXERCISE

**NAME:** TOMASZ MIAZGA

**INDEX NUMBER:** 293071

**GROUP:** 103

## 1. FILES

The program consists of two files: "Lab1.cpp" and "Sequence.h". The first one is used to perform test functions which are declared alongside with Sequence class and in "Sequence.h" file. In addition this file contains shuffle and [checkShuffleVariables](#) functions.

## 2. SEQUENCE CLASS

The class itself consists of two sections: private and public. In the private part a structure called Node is declared. It is used to "store" the linked list. In addition [Node](#) type pointer called [head](#) is declared (it is used as beginning to the list).

private:

```
struct Node {  
    Key key;  
    Info info;  
    Node* nextNode = nullptr;  
};  
  
Node* head = nullptr;
```

Public section contains the methods used in operating on the linked lists:

Methods:

```
Sequence();  
Sequence(const Sequence<Key, Info>&);  
~Sequence();  
void addNodeFront(Key, Info)  
void addNodeFixedPosition(Key, Info, int);  
void addNodeTail(Key, Info);  
void displayList();  
int returnListSize();  
void clearList();  
void removeHeadNode();  
void removeFixedNode(int);  
void removeTailNode();
```

```
Sequence<Key, Info> &operator=(const Sequence<Key, Info>&);  
Sequence<Key, Info> &operator+=(const Sequence<Key, Info>&);
```

```
void createRandomNodes(int);  
void giveNodes(const Sequence<Key, Info> &, int, int);  
bool checkExistingNodesKeys(Key);
```

Most extraordinary methods description:

- **giveNodes**: it is used exclusively in shuffle function. Its purpose is to add nodes described by start and length integers to the Sequence object which is returned by shuffle function
- **createRandomNodes**: method created only for testing purposes. Only in case of template<int,int> it creates a given number of nodes with random key and info values. If any different typenames than int are used, the method will return nothing.
- **checkExistingNodesKeys**: it searches for nodes with equivalent key as passed to the method. It is used in all node adding methods in order not to create two nodes with equal key.

### 3. TESTS

This section contains tests functions which in the code are written right under the Sequence class in "Sequence.h" file. They are designed to put to test every Sequence class method. All of them begin in creating a short linked list and using particular methods in different cases which might prove their incorrectness.

Tests:

```
void displayingListTest();  
void addDeleteNodes();  
void increaseOperatorTest();  
void equalOperatorTest();  
void copyConstructorTest();  
void listSizeTest();  
void shuffleFunctionTest();  
void addFixedPositionNodesTest();  
void checkingMethodsKeysTest();
```

What needs mentioning is that few functions test numerous methods at the same time. For example `addDeleteNodes` processes through all kinds of adding and deleting methods in various cases. In contrary, exemplary `listSizeTest` examines only one `returnListSize` method.

#### 4. SHUFFLE FUNCTION

It is a function declared outside the Sequence class in Sequence.h file. It is definitely most developed function in the whole project. It itself uses `giveNodes` method and `checkShuffleVariables` function which respectively use another methods.