

პროგრამირების აბსტრაქციები

სემინარის ამოცანები #20

string-ის გაკეთება ბმული სიით

დაწერეთ კლასი, რომელიც string-ს შეინახავს **char**-ების ბმული სიის სახით. თქვენს კლასის ობიექტის შექმნა შესაძლებელი უნდა იყოს ჩვეულებრივი სტრინგის საშუალებით. ასევე თქვენი კლასის ობიექტს უნდა შეეძლოს შესაბამისი ჩვეულებრივი string-ის დბრუნება და ორი სტრინგის კონკატენაციის ოპერაციის განხორციელება + ოპერატორის საშუალებით.

header file

```
#ifndef LL_STRING_H_
#define LL_STRING_H_

#include <string>
using namespace std;

class LLstring {
private:
    struct Node {
        char val;
        Node* next;
    };
    Node* head;
    Node* fullCopy(Node* src, Node* tail);
public:
    LLstring(string other = "");
    LLstring(const LLstring& other);
    ~LLstring();
    string toString();
    LLstring operator+ (LLstring& other);
};

#endif
```

source file

```
#include "error.h"
#include "LLstring.h"

LLstring::LLstring(string other) {
    head = NULL;
    for (int i = other.length() - 1; i >= 0; i--) {
        Node* tmp = new Node;
        tmp -> val = other[i];
        tmp -> next = head;
        head = tmp;
    }
}

LLstring::Node* LLstring::fullCopy(Node* src, Node* tail) {
    if (src == NULL)
        return tail;
    Node* result = new Node;
    result -> val = src -> val;
    result -> next = fullCopy(src -> next, tail);
    return result;
}

LLstring::LLstring(const LLstring& other) {
    head = fullCopy(other.head, NULL);
}

LLstring::~~LLstring() {
    Node* ptr = head;
    while (ptr != NULL) {
        Node* tmp = ptr;
        ptr = tmp -> next;
        delete tmp;
    }
}

string LLstring::toString() {
    string result = "";
    for (Node* ptr = head; ptr != NULL; ptr = ptr -> next)
        result.append(&(ptr -> val), 1);
    return result;
}

LLstring LLstring::operator+ (LLstring& other) {
    LLstring result;
    result.head = fullCopy(other.head, NULL);
    result.head = fullCopy(head, result.head);
    return result;
}
```