

პროგრამირების აბსტრაქციები

სემინარის ამოცანები #3

1 რიცხვების მიმდევრობის შებრუნება

დაწერეთ ფუნქცია რომელიც მომხმარებელს მიცემს საშუალებას კონსოლიდან შეიყვანოს რიცხვები, სანამ მომხმარებელი არ შეიყვანს sentinel მნიშვნელობას (ამ შემთხვევაში -1-ს). ამის შემდეგ ფუნქციამ რიცხვები უნდა დაბეჭდოს კონსოლში შებრუნებული მიმდევრობით.

```
#include <iostream>
#include "console.h"
#include "simpio.h"
#include "stack.h"
using namespace std;

const int sentinel = -1;

void printInReverse() {
    Stack<int> s;

    int currentInt = getInteger();
    while (currentInt != sentinel) {
        s.push(currentInt);
        currentInt = getInteger();
    }

    while (!s.isEmpty()) {
        currentInt = s.pop();
        cout << currentInt << " ";
    }
    cout << endl;
}
```

2 ფრჩხილებიანი გამოსახულების გამოთვლა

დაწერეთ ფუნქცია, რომელსაც გადაეცემა სტრინგში ჩაწერილი გამოსახულება და ეს ფუნქცია ამ გამოსახულების მნიშვნელობას გამოთვლის. გამოსახულება შედგება ერთნიშნა რიცხვებით, ფრჩხილებით და ერთი ტიპის არითმეტიკული ოპერაციით. ჩვენს შემთხვევაში ეს ოპერაცია გამოკლება (-) იქნება, მაგრამ მარტივად უნდა შეგეძლოთ პროგრამის გადაკეთება სხვა ოპერაციაზე. გამოსახულება სწორად იქნება შედგენილი და ამის შემოწმებაზე შეგიძლიათ არ იზრუნოთ. ასევე, სიმარტივისთვის გარანტირებულია რომ გამოსახულების დაყოველი გახსნილი ფრჩხილის დასაწყისში ჯერ უბრალოდ რიცხვი შეგხვდებათ და მხოლოდ შემდეგ შეიძლება შემდეგი ფრჩხილი გაიხსნას.

```
int evaluateExpression(string s) {
    Stack<int> savedValues;
    bool firstInteger = true;
    int currentValue;

    for (int i = 0; i < s.length(); i++) {
        if (s[i] == '-') {

        } else if (s[i] == '(') {
            savedValues.push(currentValue);
            firstInteger = true;
        } else if (s[i] == ')') {
            currentValue = savedValues.pop() - currentValue;
        } else {
            if (firstInteger) {
                currentValue = (s[i] - '0');
                firstInteger = false;
            } else {
                currentValue -= (s[i] - '0');
            }
        }
    }

    return currentValue;
}
```

3 ფიბონაჩის რიცხვები

ა) ფიბონაჩის მიმდევრობას განმარტავენ როგორც მიმდევრობას რომელშიც $F_0 = 0$, $F_1 = 1$ და მიმდევრობის ყოველი შემდეგი წევრი წინა ორი წევრის ჯამის ტოლია. დაწერეთ რეკურსიული ფუნქცია, რომელსაც გადავცემთ n -ს და გამოგვითვლის ფიბონაჩის მიმდევრობის n -ურ წევრს.

```
int fib(int n) {  
    if (1 <= n)  
        return n;  
  
    return fib(n - 2) + fib(n - 1);  
}
```

ბ) ზემოთ მოყვანილი კოდი ძალიან არაეფექტურია, იმიტომ რომ ერთიდაიგივე მნიშვნელობებს ბევრჯერ ვითვლით ცალცაკლე. ამის გამოსწორების რამდენიმე გზა არსებობს, მაგრამ ჩვენ განვიხილოთ ამოცანის შემდეგი მოდიფიკაცია: როდესაც n გადმოგვეცემა, ფუნქციამ დააბრუნოს არა მხოლოდ F_n , არამედ სტრუქტურა, რომელიც F_n -ს და F_{n+1} -ს შეიცავს. ამ ფუნქციის რეკურსიული დაწერა შეგვიძლია ისე რომ ის თავის თავს მხოლოდ ერთხელ იძახებდეს და შესაბამისად ზედმეტი გამოთვლების რაოდენობა ცოტა იყოს. დაწერეთ რეკურსიული ფუნქცია რომელიც ორ ფიბონაჩის რიცხვს აბრუნებს, და გადაწერეთ ფიბონაჩის რიცხვის გამომთვლელი ფუნქცია ისე რომ ამ ფუნქციას იყენებდეს და შესაბამისად სწრაფად მუშაობდეს.

```
struct intPair {  
    int first;  
    int second;  
};  
  
intPair fibPair(int n) {  
    if (n == 0) {  
        intPair result;  
        result.first = 0;  
        result.second = 1;  
        return result;  
    }  
  
    intPair previous = fibPair(n-1);  
    intPair result;  
    result.first = previous.second;  
    result.second = previous.first + previous.second;  
    return result;  
}  
  
int fib(int n) {  
    intPair result = fibPair(n);  
    return result.first;  
}
```