

პროგრამირების აბსტრაქციები

სემინარის ამოცანები #1

1 რიცხვების მინიმუმი, მაქსიმუმი და საშუალო

დაწერეთ პროგრამა, რომელიც მომხმარებელს მოსთხოვს კონსოლიდან მთელი რიცხვების შეყვანას და ბოლოს გამოიტანს ამ რიცხვებიდან მინიმუმს, მაქსიმუმს და საშუალოს.

პროგრამამ მომხმარებელს რიცხვების შეყვანა უნდა სთხოვოს სათითაოდ, სანამ მომხმარებელი არ შეიყვანს სპეციალურ რიცხვს (ე.წ. სენტინელს) რომელიც მიმდევრობის დასასრულს აღნიშნავს. სენტინელად (-1) აიღეთ, მაგრამ თქვენი პროგრამის გადაკეთება მარტივად უნდა იყოს შესაძლებელი სენტინელის სხვა მნიშვნელობებზე.

```
#include <iostream>
#include "console.h"
#include "simpio.h"

using namespace std;

const int sentinel = -1;

struct statsT {
    int min;
    int max;
    double avg;
};

statsT getSequenceStats(int sentinel) {
    int count = 0;
    int sum = 0;
    statsT result;
    result.min = INT_MAX;
    result.max = INT_MIN;
    while (true) {
        cout << "Enter next integer (or " << sentinel << " to terminate): ";
        int input = getInteger();
        if (input == sentinel) break;
        if (input > result.max) result.max = input;
        if (input < result.min) result.min = input;
        sum += input;
        count += 1;
    }
    result.avg = double(sum)/count;
    return result;
}

int main() {
    statsT ans = getSequenceStats(sentinel);
    cout << "min = " << ans.min << endl;
    cout << "max = " << ans.max << endl;
    cout << "avg = " << ans.avg << endl;
    return 0;
}
```

2 2-ის ხარისხია თუ არა რიცხვი

დაწერეთ რეკურსიული ფუნქცია, რომელსაც გადაეცემა ერთი დადებითი მთელი რიცხვი და ფუნქცია შეამოწმებს ეს მთელი რიცხვი ორის ხარისხი არის თუ არა. (მინიშნება: მთელი რიცხვი ორის ხარისხია თუ ის ან 1-ის ტოლია, ან ლუწია და მისი ორზე გაყოფის შედეგად ორის ხარისხია)

```
bool isPowerOfTwo(int n) {  
    if (n % 2 == 1)  
        return (n == 1);  
  
    return isPowerOfTwo(n/2);  
}
```

3 ევკლიდეს ალგორითმი

დაწერეთ პროგრამა, რომელიც მომხმარებელს მოსთხოვს ორი მთელი დადებითი რიცხვის შეყვანას, შეამოწმებს რომ შეყვანილი რიცხვები დადებითია და გამოთვლის (და დაბეჭდავს) ამ რიცხვების უდიდეს საერთო გამყოფს. უ.ს.გ. რეკურსიულად უნდა გამოთვალოთ ევკლიდეს ალგორითმის გამოყენებით.

```
#include <iostream>  
#include "console.h"  
#include "simpio.h"  
  
using namespace std;  
  
int gcd(int, int);  
  
int main() {  
    cout << "Enter two positive integers" << endl;  
    int x = getInteger();  
    while (x <= 0) {  
        cout << "Please make sure the integer is positive" << endl;  
        x = getInteger();  
    }  
    int y = getInteger();  
    while (y <= 0) {  
        cout << "Please make sure the integer is positive" << endl;  
        y = getInteger();  
    }  
  
    int g = gcd(x, y);  
    cout << "gcd is: " << g << endl;  
    return 0;  
}  
  
int gcd(int a, int b) {  
    if (b == 0)  
        return a;  
  
    return gcd(b, a % b);  
}
```