

АНАЛИЗ И РЕШЕНИЕ НА ЗАДАЧА TRICK

Единственото предизвикателство в задачата е свързано с липсата на конкретно ограничение за обема на входните данни.

Очевидно е добре да се използват две линейни структури от данни – една за числата и една – за операциите и след прочитане на данните, да се обработят в обратен ред. Масивът е добро решение, но предвид, че няма ограничение за броя на операциите и операндите, по-удачно би било да се използва линейна динамична структурата. Като добавим съображението за обработка в обратен ред, най-удачно се оказва използването на стек.

```
//trick.cpp

#include <iostream>
#include <stack>

using namespace std;

stack<char> op;           //operations
stack<int> arg;          //arguments

int res;                 //result
void read();            //inputs data into both stacks
                        //the number after '=' is stored in
                        //the res variable
void calc();            //calculates the initial number
                        //going in reverse order
                        //and using the opposite operations
                        //The result is kept in the res
                        variable

int main()
{
    read();
    calc();
    cout << res << endl;

    return 0;
}

void read(){
    char o;
    int a;

    while ((cin >> o >> a) && (o != '=')){
        op.push(o);
        arg.push(a);
    }
    res = a;
}

void calc(){
    int a;
    char o;

    while (!arg.empty()){
        a = arg.top();
        arg.pop();
        o = op.top();
        op.pop();
        switch(o){
            case '+': res -= a;
                break;
            case '-': res += a;
                break;
            case '*': res /= a;
                break;
        }
    }
}
```

Автор: Николина Николова