



Назив проблема: Пронађи

Временско ограничење: 1 секунда
Меморијско ограничење: 64MB

Текст проблема

Перица и Марица су открили нову игру. Марица замисли неки сортирани низ бројева и каже Перици неки број из тог низа. Перичин задатак је да погоди прву позицију тог броја у Марицином низу. Да би то успео, он може Марици да поставља следећу врсту упита:

Да ли је број у низу на позицији Idx већи, мањи или једнак од броја X ?

Помозите Перици да погоди позицију замишљеног броја.

Описи функција

Потребно је да имплементирате следеће две функције:

1. $Init(N)$ – ова функција се позива само једном на почетку и означава да је Марица замислила сортирани низ дужине N . Можете је искористити да поставите почетне вредности својих глобалних променљивих/низова.
2. $Solve(X)$ – ова функција се позива једном и треба да врати прво појављивање броја X у Марицином низу. Овде можете да позивате функцију $Compare(idx, x)$ која враћа:
 - 1 ако је елемент у Марицином низу на позицији idx већи од x
 - -1 ако је елемент у Марицином низу на позицији idx мањи од x
 - 0 ако је елемент у Марицином низу на позицији idx једнак x

Обратите пажњу да је Марица **индексирала њен низ од 1 до N** . Марицин низ је **сортиран**.

Пример:

Нека је $N = 5$ и нека је Марица замислила низ $A = [1, 5, 8, 10, 12]$ и нека је Перици рекла број $X = 10$. Пошто се $X = 10$ налази на позицији 4 у Марицином низу, функција $Solve(10)$ треба да врати 4.

Ограничења

- $1 \leq N \leq 100.000$
- $1 \leq A[i] \leq 1.000.000$, низ A је сортиран
- $1 \leq X \leq 1.000.000$ и број X се налази у низу A

Подзадачи

- **ПОДЗАДАТАК 1 [25 ПОЕНА]:** Сви елементи низа ће бити различити. Дозвољено је позвати функцију $Compare(idx, x)$ највише 100.000 пута
- **ПОДЗАДАТАК 2 [25 ПОЕНА]:** Сви елементи низа ће бити различити. Дозвољено је позвати функцију $Compare(idx, x)$ највише 20 пута



- **ПОДЗАДАТАК 3 [25 ПОЕНА]:** Дозвољено је позвати функцију $Compare(idx, x)$ највише 100.000 пута
- **ПОДЗАДАТАК 4 [25 ПОЕНА]:** Дозвољено је позвати функцију $Compare(idx, x)$ највише 20 пута

Детаљи имплементације

Потребно је да пошаљете тачно један фајл, под називом **pronadji.c**, **pronadji.cpp** или **pronadji.pas**, који имплементира горе поменуте функцију. Осим тражених функција, ваш фајл може садржати и додатне глобалне променљиве, помоћне функције и додатне библиотеке. Уколико радите у C/C++-у, потребно је на почетку фајла ставити **#include "pronadji.h"** а уколико радите у Pascal-у, потребно је на почетку фајла ставити:

unit pronadji;

interface

procedure Init(M : LongInt);

function Solve(x : LongInt) : LongInt;

implementation

uses graderhelplib;

(ово је већ додато у фајловима који су вам обезбеђени).

Зависно од програмског језика који користите, ваша функција/процедура мора бити следећег облика:

C/C++	<code>void Init(int M); int Solve(int x);</code>
Pascal	<code>procedure Init(M : LongInt); function Solve(x : LongInt) : LongInt;</code>

На располагању вам је функција $Compare(idx, x)$

C/C++	<code>int Compare(int idx, int x);</code>
Pascal	<code>function Compare(idx, x : LongInt) : LongInt;</code>

Тестирање и експериментисање

Уз задатак, обезбеђени су вам "template" фајлови (pronadji.c, pronadji.cpp, pronadji.pas) које можете користити и мењати по потреби. Такође су вам обезбеђени програми (grader.c, grader.cpp, grader.pas, pronadji.h, graderhelplib.pas) који служе да лакше тестирате кодове. Ови програми учитавају са стандардног улаза следеће податке:



-
- У првом реду бројеве N, Idx и $Calls$, раздвојене размаком,
 - У другом реду N бројева A_i , раздвојених размаком

Где је N број елемената Маричиног низа, Idx позиција замичљеног елемента у низу и $Calls$ број дозвољених позива функције *Compare*.