

Открий числото

Ави и Боби играят следната игра: Ави си е намисля редица от N числа a_0, a_1, \dots, a_{N-1} , подредени в строго нарастващ ред, т.е. $a_i < a_{i+1}$ за всяко $0 \leq i \leq N-2$. След това тя избира едно число от редицата a_k и го съобщава на Боби. Боби трябва да намери индекса на числото (т.е. k) като може да задава на Ави въпроси от вида „Каква е стойността на a_i ?“ за избрана от него стойност i .

Вашата задача е да напишете програма, играеща ролята на Боби, която да намира индекса k на избраното от Ави число a_k , задавайки най-много Q въпроса.

Детайли по имплементацията

Вашата програма трябва да съдържа в началото си `#include "findnumber.h"`.
Вашата програма трябва да реализира следната функция:

- `int find_index(int N, int a)`
 - N : дължината на намислената от Ави редица
 - a : стойността на избраното от Ави число a_k
 - Функцията трябва да връща стойността на индекса k .
 - Функцията се вика веднъж за всеки тест.

Вашата програма може да задава въпроси, използвайки следната функция:

- `int get_number(int i)`
 - i : индексът на числото, за което Вашата програма пита. i трябва да е валиден индекс, т.е. $0 \leq i \leq N-1$.
 - Функцията връща стойността на a_i .
 - Функцията може да се вика Q пъти за един тест.

В случай, че наруши ограниченията за параметъра на `get_number`, ще получите съобщение „Invalid question asked.“.

В случай, че извикате функцията `get_number` повече от Q пъти, ще получите съобщение „Too many questions asked.“.

Пример

Грейдърът извършва следното извикване на функция:

- `find_index(5, 3)` Ави е намислила редица с $N = 5$ елемента и Боби търси индекса на числото със стойност 3.

Програмата извършва следните извиквания на функции:

- `get_number(0)` връща 0
- `get_number(3)` връща 7
- `get_number(1)` връща 1
- `get_number(2)` връща 3
- `get_number(4)` връща 5

Редицата на Ави е 0, 1, 3, 5, 7 и индексът на търсената стойност 3 е 2. Съответно `find_index` трябва да върне 2.

Подзадачи

Във всички подзадачи $0 \leq a_i \leq 10^9$ за всяко $0 \leq i \leq N - 1$.

Нека означим с Q максималния разрешен брой извиквания на `get_number` за едно изпълнение на програмата.

1. (50 точки) $1 \leq N \leq 100\,000$, $Q = 100\,000$
2. (50 точки) $1 \leq N \leq 100\,000$, $Q = 17$

Локално тестване

За да можете да тествате решението си на компютъра си, Ви се предоставят файловете `Lgrader.cpp` и `findnumber.h`, които да компилирате заедно с Вашето решение `findnumber.cpp`.

Вход

- Ред 1: три цели числа N , k и Q
- Ред 2: N числа a_0, a_1, \dots, a_{N-1}

Изход

- Ред 1:
 - „*Output is correct.*“, ако програмата е преминала успешно теста.
 - „*Invalid question.*“, ако програмата е задала въпрос за стойността на невалиден индекс i .
 - „*Too many questions asked.*“, ако програмата е извършила повече от Q извиквания на `get_number`.
 - „*Output isn't correct.*“, ако програмата не е надвишила максималния разрешен брой въпроси, но не е намерила правилно търсения индекс k .

Изпращане на тестове към системата

Можете да изпращате собствени тестове към системата. Форматът на входните и изходните данни е същият като този на предоставения локален грейдър.