
Arches**X44670_en**

A sequence x_1, x_2, \dots, x_n of integers of length $n \geq 2$ is an *arch* if the following conditions are satisfied:

- The first and the last element of the sequence are 0.
- All elements, other than the first and the last, are strictly positive, that is, for all i , $1 < i < n$, we must have $x_i > 0$.
- The difference between any two consecutive elements in the sequence is +1, 0, or -1, that is, for all i , $1 < i \leq n$, we must have $x_i - x_{i-1} \in \{-1, 0, +1\}$

Consider a sequence of concatenated archs like this one

0 1 2 1 0 1 2 2 2 3 4 3 2 3 2 1 0 0

Notice that the ending 0 of an arch is the initial 0 of the next arch. The example sequence contains 3 arches. The first arch is 0 1 2 1 0, the second arch is 0 1 2 2 2 3 4 3 2 3 2 1 0 and the last arch is 0 0.

Write a function `int count_initial_arches()` that reads a sequence of integers from the standard input (`cin`) and returns the number of arches found before we encounter a non-arch.

For instance, for a sequence that starts like this:

0 1 2 1 0 1 0 0 2 2 2 3 4 3 2 3 2 1 0 0 ...

the answer is 3; we find the three arches 0 1 2 1 0, 0 1 0, and 0 0, then followed the subsequence 0 2 2 3 ... which is not an arch.

Use the following C++ structure. Note that in general it is not necessary to read the whole input to obtain the answer, and not taking this into account could give you a "Time Limit Exceeded" verdict.

```
#include <iostream>

using namespace std;

// Add here other function definitions if you need them.

// Read problem statement.
int count_initial_arches()
{
    // Write here your code
    ...
}

int main()
{
    cout<<count_initial_arches()<<endl;
}
```

=====

CASTELLANO

=====

Una secuencia x_1, x_2, \dots, x_n de enteros de longitud $n \geq 2$ es un *arco* si satisface las siguientes condiciones:

- El primer y el último elementos de la secuencia son 0.
- Todos los elementos, exceptuados el primero y último, son estrictamente positivos, esto es, para todo $i, 1 < i < n$, se cumple que $x_i > 0$.
- La diferencia entre cualesquiera dos elementos consecutivos de la secuencia es +1, 0 ó -1, es decir, para todo $i, 1 < i \leq n$, se cumple que $x_i - x_{i-1} \in \{-1, 0, +1\}$.

Considera ahora una secuencia de arcos concatenados como la siguiente:

0 1 2 1 0 1 2 2 2 3 4 3 2 3 2 1 0 0

Observa que el 0 al final de un arco es el 0 inicial del siguiente arco. La secuencia del ejemplo contiene 3 arcos. El primer arco es 0 1 2 1 0, el segundo arco es 0 1 2 2 2 3 4 3 2 3 2 1 0 y el último arco es 0 0.

Escribe una función `int count_initial_arches()` que lee una secuencia de enteros de la entrada estándar (`cin`) y retorna el número de arcos encontrado antes de encontrar un no-arco. Por ejemplo, para una secuencia que comienza de la siguiente manera:

0 1 2 1 0 1 0 0 2 2 2 3 4 3 2 3 2 1 0 0 ...

la respuesta es 3; hallamos los tres arcos 0 1 2 1 0, 0 1 0, y 0 0, y a continuación la subsecuencia 0 2 2 3 ... que no es un arco.

Usa el código C++ dado más arriba. Observa que en general no es necesario leer toda la entrada para obtener la respuesta, y no tener esto en cuenta puede conducir a un veredicto "Time Limit Exceeded".

=====

CATALÀ

=====

Una seqüència x_1, x_2, \dots, x_n d'enters de longitud $n \geq 2$ és un *arc* si satisfà les següents condicions:

- El primer i l'últim elements de la seqüència són 0.
- Tots els elements, exceptuats el primer i últim, són estrictament positius, és a dir, per a tot $i, 1 < i < n$, es compleix que $x_i > 0$.
- La diferència entre qualssevol dos elements consecutius de la seqüència és +1, 0 o -1, és a dir, per a tot $i, 1 < i \leq n$, es compleix que $x_i - x_{i-1} \in \{-1, 0, +1\}$.

Considera ara una seqüència d'arcs concatenats com la següent:

0 1 2 1 0 1 2 2 2 3 4 3 2 3 2 1 0 0

Observa que el 0 al final d'un arc és el 0 inicial del següent arc. La seqüència de l'exemple conté 3 arcs. El primer arc és 0 1 2 1 0, el segon arc és 0 1 2 2 2 3 4 3 2 3 2 1 0 i l'últim arc és 0 0.

Escriu una funció `int count_initial_arches()` que llegeix una seqüència d'enters de l'entrada estàndard (`cin`) i retorna el nombre d'arcs trobat abans de trobar un no-arc. Per exemple, per a una seqüència que comença de la següent manera:

0 1 2 1 0 1 0 0 2 2 2 3 4 3 2 3 2 1 0 0 ...

la resposta és 3; trobem els tres arcs 0 1 2 1 0, 0 1 0, i 0 0, i tot seguit la subseqüència 0 2 2 3 ... que no és un arc.

Utilitza el codi C ++ donat més amunt. Observa que en general no cal llegir tota l'entrada per obtenir la resposta, i no tenir això en compte pot conduir a un veredict "Time Limit Exceeded".

Exam score: 2.5 **Automatic part:** 30%

Input

A sequence of integers.

=====

CASTELLANO

=====

Una secuencia de enteros.

=====

CATALÀ

=====

Una seqüència d'enters.

Output

The program prints in the cout the number of arches read until a non-arch occurs in the input.

=====

CASTELLANO

=====

El programa imprime en la salida (cout) el número de arcos leídos hasta el primer no-arco en la entrada.

=====

CATALÀ

=====

El programa imprimeix a la sortida (cout) el nombre d'arcs llegits fins al primer no-arc a l'entrada.

Sample input 1

0 0 0 0 1 1 0 0 1 2 4 3 2 1 0 0 1 0

Sample output 1

5

Sample input 2

1 0 1 2 1 2 2 3 4 5 5 6 5 4 5 4 4 4 4 3 2

Sample output 2

0 0 0

Sample input 3

0 1 0 1 2 1 2 2 3 4 5 5 6 5 4 5 4 4 4 4 3

Sample output 3

2 1 0

Sample input 4

0 1 2 2 1 1 0 1 0 -1 0 1 2 1 0 2 0 1 0

Sample output 4

2

Problem information

Author : PRO1

Generation : 2016-05-08 14:45:17

© *Jutge.org*, 2006–2016.

<http://www.jutge.org>