

Problem 1

Inspired by the Dire Straits song, "Sultans of Swing", you have decided to become a rock star. After a lengthy recording session, you have acquired:

- a total of **ff** songs that start fast and end fast,
- a total of **fs** songs that start fast and end slow,
- a total of **sf** songs that start slow and end fast, and
- a total of **ss** songs that start slow and end slow.

It remains only to determine which of these songs should go on your first album, and in what order they should appear. Of course, no song can appear on the album more than once.

Unfortunately, your record company has placed several restrictions on your album:

1. A song that ends fast may only be immediately followed by a song that starts fast.
2. A song that ends slow may only be immediately followed by a song that starts slow.
3. If you have at least one song that starts fast, then the first song on the album must start fast. Otherwise, this restriction can be ignored.

At this stage in your artistic career, you must do what your record company has ordered, but you do want to place as many songs as possible on the album.

Given ints **ff**, **fs**, **sf**, and **ss**, representing the quantities described above, return the maximum number of songs that can be placed on a single album without violating the record company's restrictions.

***Input data**

The input file `songs.in` will have four lines containing one number, each representing the number of `ff`, `fs`, `sf` and `ss` songs (in this order and with the significance given in the problem description).

***Output data**

The output file `songs.out` should contain a single line with the integer representing the maximum number of songs that can be placed on a single album.

***Restrictions**

ff, **fs**, **sf**, and **ss** will each be between 0 and 1000 inclusive.

At least one of **ff**, **fs**, **sf** or **ss** will be greater than 0.

***Example**

<code>songs.in</code>	<code>songs.out</code>
100	100
0	
0	
200	
0	201
0	
20	
200	
192	999
279	
971	
249	

Problem 2

The National Bank publishes the EUR-RON currency many times during one day and these reports show all the fluctuations which occur during the day. For the majority of the people this detailed fluctuation is not necessary during small periods of time. Thus, The National Bank wishes to publish the currency fluctuation in a way in which it shows that the rate has grown up and then fell.

Compute the maximum number of currency rates (CR) which can be chosen for publishing, such that the people understand that in that day the currency has grown and then fell. The currency rates chosen for publishing must be in chronological order.

***Input data**

The input file `currency.in` is structured as follows:

- first line contains an integer N, which represents the number of currency rates(CR)
- on each of the following N lines there is one currency value; the currency rates are in chronological order.

***Output data**

The output file `currency.out` should contain a single line with the integer representing maximum number of currency rates which can be chosen for publishing.

***Restrictions**

$2 < N \leq 1000$

$4.0000 \leq CR \leq 5.0000$

All CRs are different.

The CRs chosen must have an increasing sequence (with at least two elements), followed by a decreasing sequence (with at least one element).

***Example**

<code>currency.in</code>	<code>currency.out</code>
--------------------------	---------------------------

12	9
4.1120	
4.2470	
4.2110	
4.2090	
4.2440	
4.3500	
4.4700	
4.5100	
4.3120	
4.2150	
4.1170	
4.2170	

Explanation

CRs chosen are: 4.1120, 4.2110, 4.2440, 4.3500, 4.4700, 4.5100, 4.3120, 4.2150, 4.1170

Note:

The problems were taken from:

<http://www.topcoder.com/>

<http://infoarena.ro/>

The solutions should have a Readme file that should contain:

1. a short description of the algorithms you used,
2. the complexity of the algorithms (you must compute it).

Send the solutions in a .zip archive with the name `HW_<number>_<name>_<group>.zip` (e.g. `HW_1_PopescuAndrei_1231E.zip`) by email to: andavintila@gmail.com .

The deadline for receiving the homework is 5th of December, at 23:59.

Rules for assignments: <http://adcfils.wordpress.com/assignments/>