

Colombian Collegiate Programming League

CCPL 2022

Universidad de Medellín (test)

Problems

This set contains 10 problems; pages 1 to 4.

(Borrowed from several sources online.)

A - Bender B. Rodríguez Problem	1
B - Tennis Championship	3
C - 2 the 9's	4

Official site <http://programmingleague.org>

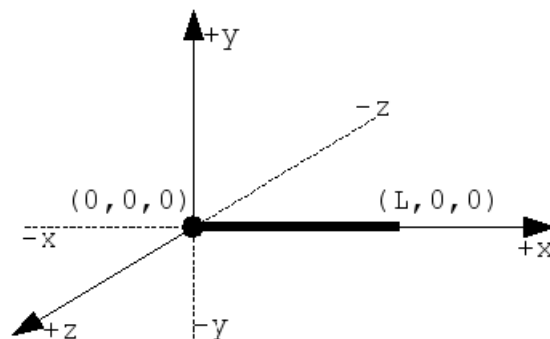
Follow us on Twitter @CCPL2003

A - Bender B. Rodríguez Problem

Source file name: `bender.c`, `bender.cpp`, `bender.java`, or `bender.py`

Bender is a robot built by *Mom's Friendly Robot Company* at its plant in Tijuana, Mexico in 1996. He is a Bending-Unit 22, serial number 2716057 and chassis number 1729. He was created for the task of bending metal wires.

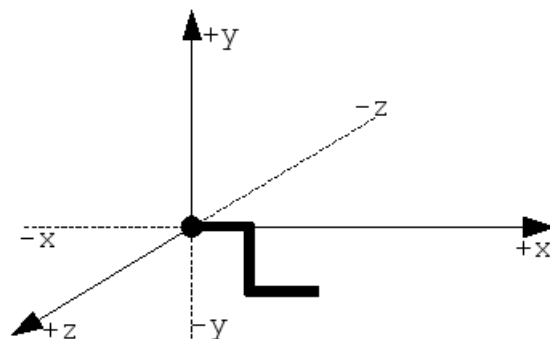
Bender needs to bend a wire of length L ($L \geq 2$ an integer). The wire is represented in the Bender's brain (a MOS Technology 6502 microprocessor) as a line stuck in the origin of a tridimensional cartesian coordinate system, and extended along the x positive axis ($x+$), so that the fixed extreme of the wire is in the coordinate $(0, 0, 0)$ and the free extreme of the wire is in the coordinate $(L, 0, 0)$.



Bender bends the wire at specific points, starting at the point $(L-1, 0, 0)$ and ending at the point $(1, 0, 0)$. For each i from $L-1$ to 1 , Bender can take one of the following decisions:

- Not to bend the wire at point $(i, 0, 0)$.
- To bend the wire at point $(i, 0, 0)$ an angle of $\frac{\pi}{2}$ to be parallel to the axis $+y$, $-y$, $+z$ or $-z$.

For example, if $L=3$ and Bender bends the wire at $(2, 0, 0)$ on the $+y$ axis direction, and at $(1, 0, 0)$ on the $-y$ axis direction, the result would be:



Given a sequence of bends, you must determine what direction is pointed by the last segment of the wire ($+x$ in the example). You can suppose that the wire can intercept itself, after all it is the future!

Input

The first line of each test case gives an integer L ($2 \leq L \leq 100000$) indicating the length of the wire.

The second line of each test case contains the $L-1$ decisions taken by Bender at each point, separated by spaces. The j -th decision in the list (for each $1 \leq j \leq L-1$) corresponds to the decision taken at the point $(L-j, 0, 0)$, and must be one of the following:

- No if the wire isn't bended at point $(L-j, 0, 0)$.
- +y if the wire is bended at point $(L-j, 0, 0)$ on the +y axis.
- -y if the wire is bended at point $(L-j, 0, 0)$ on the -y axis.
- +z if the wire is bended at point $(L-j, 0, 0)$ on the +z axis.
- -z if the wire is bended at point $(L-j, 0, 0)$ on the -z axis.

The end of the input is indicated when $L=0$.

The input must be read from standard input.

Output

For each case in the input, print one line with the direction pointed by the last segment of the wire, +x, -x, +y, -y, +z or -z depending on the case. *The output must be written to standard output.*

Sample input	Output for the sample input
3	+x
+z -z	+z
3	+z
+z +y	-x
2	+z
+z	
4	
+z +y +z	
5	
No +z No No	
0	

B - Tennis Championship

Source file name: `champion.c`, `champion.cpp`, `champion.java`, or `champion.py`

A certain tennis championship with P players has a particular set of rules:

1. Before every round, players are paired randomly.
2. Each pair so defined establishes a match that will be played.
3. The winner of a match advances to the next round in the tournament and the loser is eliminated from competition.
4. If the number of players before a round is odd, then one player (chosen at random) is automatically promoted to the next round.

This process should be repeated over and over again until there is exactly one player left. Such a player will be the champion.

The Tennis Championship Organization wants to calculate the total number of matches needed to determine the champion.

Input

The input consists of several test cases, each one consisting of a single line containing a positive integer P , the number of players.

The input must be read from standard input.

Output

For each test case, output a line with one integer indicating the number of matches needed to determine the champion.

The output must be written to standard output.

Sample Input	Sample Output
3	2
2	1

