ACM International Collegiate Programming Contest
2016

Latin American Regional Contests

November 11th-12th, 2016

Warmup Session

This problem set contains 3 problems; pages are numbered from 1 to 3.

This problem set is used in simultaneous contests hosted in the following countries:

Argentina, Bolivia, Brasil, Chile, Colombia, Costa Rica, Cuba,
México, Panamá, Perú, República Dominicana and Venezuela

General information

Unless otherwise stated, the following conditions hold for all problems.

Program name

1. Your solution must be called *codename*`.c`, *codename*`.cpp` or *codename*`.java`, *codename*`.py2`, *codename*`.py3`, where *codename* is the capital letter which identifies the problem.

Input

1. The input must be read from standard input.
2. The input consists of a single test case, which is described using a number of lines that depends on the problem. No extra data appear in the input.
3. When a line of data contains several values, they are separated by *single* spaces. No other spaces appear in the input. There are no empty lines.
4. The English alphabet is used. There are no letters with tildes, accents, diaereses or other diacritical marks (ñ, Ã, é, Ì, ô, Ü, ç, etcetera).
5. Every line, including the last one, has the usual end-of-line mark.

Output

1. The output must be written to standard output.
2. The result of the test case must appear in the output using a number of lines that depends on the problem. No extra data should appear in the output.
3. When a line of results contains several values, they must be separated by *single* spaces. No other spaces should appear in the output. There should be no empty lines.
4. The English alphabet must be used. There should be no letters with tildes, accents, diaereses or other diacritical marks (ñ, Ã, é, Ì, ô, Ü, ç, etcetera).
5. Every line, including the last one, must have the usual end-of-line mark.
6. To output real numbers, round them to the closest rational with the required number of digits after the decimal point. Test case is such that there are no ties when rounding as specified.

Development team

The following persons helped to develop the problem set by creating and improving statements, solutions, test cases and input and output checkers:

Bruno Junqueira Adami, Brazil

Fidel I. Schaposnik, Argentina

Guilherme Albuquerque Pinto, Brazil

Luis Fernando Dorelli de Abreu, Brazil

Paulo Cezar Pereira Costa, Brazil

Rafael Armando Garcia Gomez, Colombia

Yonny Mondelo Hernández, Cuba

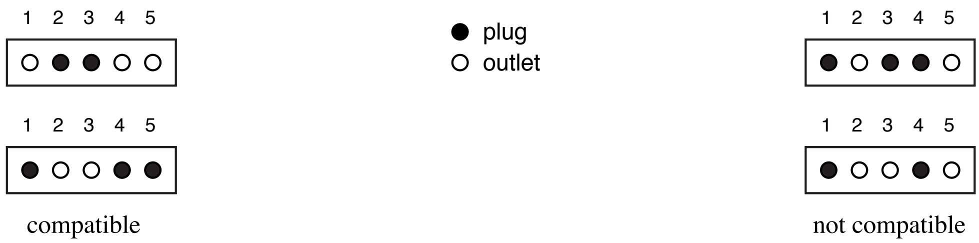
Problem A – Automated Checking Machine

Author: Ricardo Anido, Brazil

The Internet Computer Parts Company (ICPC) is an on-line shop that sells computer parts. Pairs of in-line electrical connectors are among the most popular parts that ICPC sells. However, they are also one of the parts that are returned more often by unsatisfied customers, because due to errors in packaging the connectors sent to the costumers may not be *compatible*.

An in-line connector is composed of five connection points, labelled from 1 to 5. Each connection point of a connector can be either a plug or an outlet. We say two connectors are *compatible* if, for every label, one connection point is a plug and the other connection point is an outlet (in other words, two connectors are compatible if, for every connection point with the same label, a plug and an outlet meet when the two connectors are connected).

The figure below shows examples of two connectors that are compatible and two connectors that are not compatible.



ICPC is introducing a state-of-the-art Automated Checking Machine (ACM), with an optical checker, which will verify whether the two connectors packaged for a customer are indeed compatible. The complex and expensive hardware of the ACM is ready, but they need your help to finish the software.

Given the descriptions of a pair of in-line connectors, your task is to determine if the connectors are compatible.

Input

The first line contains five integers X_i ($0 \leq X_i \leq 1$ for $i = 1, 2, \ldots, 5$), representing the connection points of the first connector in the pair. The second line contains five integers Y_i ($0 \leq Y_i \leq 1$ for $i = 1, 2, \ldots, 5$), representing the connection points of the second connector. In the input, a 0 represents an outlet an a 1 represents a plug.

Output

Output a line with a character representing whether the connectors are compatible or not. If they are compatible write the uppercase letter "`Y`"; otherwise write the uppercase letter "`N`".

Sample input 1	**Sample output 1**
`1 1 0 1 0` `0 0 1 0 1`	`Y`

Sample input 2	**Sample output 2**
`1 0 0 1 0` `1 0 1 1 0`	`N`

Problem B – Different Digits

Author: Ines Kereki, Uruguay

The inhabitants of Nlogonia are very superstitious. One of their beliefs is that street house numbers that have a repeated digit bring bad luck for the residents. Therefore, they would never live in a house which has a street number like 838 or 1004.

The Queen of Nlogonia ordered a new seaside avenue to be built, and wants to assign to the new houses only numbers without repeated digits, to avoid discomfort among her subjects. You have been appointed by Her Majesty to write a program that, given two integers N and M, determines the maximum number of houses that can be assigned street numbers between N and M, inclusive, that do not have repeated digits.

Input

The input consists of a single line that contains two integers N and M, as described above ($1 \leq N \leq M \leq 5000$).

Output

Output a line with an integer representing the number of street house numbers between N and M, inclusive, with no repeated digits.

Sample input 1	Sample output 1
87 104	14

Sample input 2	Sample output 2
989 1022	0

Sample input 3	Sample output 3
22 25	3

Sample input 4	Sample output 4
1234 1234	1

Problem C – Competition

Author: Bruno Junqueira Adami, Brazil

Bob and Alice are participating in a programming contest as a team. The contest has N problems which must be solved in order. Naturally there are some problems that they cannot solve, in that case they can skip it. There may be also problems that only Bob or Alice alone can solve.

They want to solve all the problems possible switching as few times as possible who is at the computer programming the solution.

Given the number of problems and the problems that Bob and Alice can solve, calculate the minimum number of switches between the usage of the computer. Anyone can start using it.

Input

The first line contains three integers N ($1 \leq N \leq 10^9$), A ($1 \leq A \leq min(N, 5 * 10^4)$) and B ($1 \leq B \leq min(N, 5 * 10^4)$). The next line contains A unique integers denoting the problems Alice can solve. The following line contains B unique integers denoting the problems Bob can solve. The first problem is denoted by the number 1, the second by number 2, the N-th by N, and so on.

Output

Output a line with an integer representing the minimum number of switches between the usage of the computer.

Sample input 1	**Sample output 1**

`5 2 3`
`2 4`
`1 5 3`
	`4`

Sample input 2	**Sample output 2**

`4 3 3`
`1 2 3`
`2 3 4`
	`1`

Sample input 3	**Sample output 3**

`4 3 3`
`1 3 4`
`4 3 1`
	`0`