

11th QMUL S&E Programming Competition

Sponsored by

School of Electronic Engineering and Computer Science

Wednesday, 21st February 2018

Online judge and documentation. These can be found by visiting:

<http://138.37.90.47:3333/>

Input/Output. Your programs should read from standard input. The programs should process one input and print the result at the standard output. When judging your submission we will run your program several times on different inputs.

Java 1.8. If you require java 1.8 on the terminal window (command line), make sure you run:

```
$ module load java/1.8.0_144-oracle
```

Pairing by Height

(If using Java you should name your class 'A')

Given a list of people and their respective heights, decide whether it is possible to pair them up in such a way that in each pair the difference between their heights is at most 2cm.

Input Specification

The input will consist of a positive even number n ($2 \leq n \leq 10000$) followed by n lines where in each line you have the first name of a person followed by their height in meters (with centimetres precision). All names will be guaranteed to be distinct and have at most 20 alphanumeric characters.

Output Specification

Output "Yes" or "No", depending on whether a solution can be found.

Sample Inputs.

Input	Output
6 Peter 1.50m John 1.66m Lucy 1.65m Nicky 1.51m Silvia 1.65m Maria 1.75m	No
6 Peter 1.50m John 1.77m Lucy 1.65m Nicky 1.51m Silvia 1.65m Maria 1.75m	Yes

Memorable Phone Numbers

(If using Java you should name your class 'B')

A telecommunication company has decided to charge a premium for phone numbers that are easy to remember. They have noticed that any of the following characteristics help a user to remember their number:

1. The number contains a consecutive subsequence of identical digits of length at least three, e.g. 07772345343 contains three '7's in a row.
2. The number contains a consecutive subsequence of digits in an arithmetic progression (increasing or decreasing) of length at least three, e.g. 07872466283 contains the arithmetic progression 2,4,6. You should only consider arithmetic progressions where the difference between the digits is non-zero.
3. The number contains a subsequence of length at least five which is a palindrome, e.g. 08776545622 contains the length 5 palindrome '65456'.

A number is then considered to be "easy to remember" if it has at least two of the three characteristics above.

Write a program that outputs 'Yes' or 'No' depending on whether a given phone number is memorable or not. The two matches are allowed to overlap, e.g. 07775312659 is memorable because it has a subsequence with three identical '7's but it also has the arithmetic progression 7,5,3.

Input Specification

A phone number with 11 digits (no spaces).

Output Specification

Output "Yes" or "No" depending on whether the given number is memorable or not.

Sample Inputs.

Input	Output
07775312659	Yes
07765312659	No

Buggy Libraries

(If using Java you should name your class 'C')

It is common for a software library to depend on several other libraries. The problem is that when one of the libraries contains a bug, all the other libraries that depend on it will be compromised. Given a list of software dependencies, and the name of a buggy library, your program should identify all the other libraries that will be compromised.

Input Specification

The first line of the input contains a number n ($1 \leq n \leq 200$) and the name of the buggy library. Then follow n lines, each containing the name of a library followed by a list of at most 10 libraries it depends on. There won't be any cyclic dependencies. Library names consist of a sequence of alphanumeric characters of length at most 20.

Output Specification

You program should output the sentence:

```
A bug in library XX will break libraries: YY1,YY2,...,YYk
```

where XX is the buggy library, and $YY1,YY2,\dots,YYk$ is a list of libraries that depend on XX . These should be listed in lexicographical order. If no library depends on XX your program should output instead

```
A bug in library XX will not break any other libraries.
```

Sample Inputs.

Input	Output
<pre>5 lib7 lib1 : lib2 lib3 lib2 : lib4 lib3 : lib4 lib5 lib6 lib4 : lib5 lib7 lib6 : lib5</pre>	<pre>A bug in library lib7 will break libraries: lib1,lib2,lib3,lib4</pre>
<pre>4 lib1 lib1 : lib2 lib3 lib2 : lib4 lib3 : lib4 lib5 lib6 lib4 : lib5 lib7</pre>	<pre>A bug in library lib1 will not break any other libraries.</pre>

British Coins

(If using Java you should name your class 'D'.)

Given an amount between 1p and £10, determine the smallest number of UK coins (£2, £1, 50p, 20p, 10p, 5p, 2p, 1p) needed to make up that amount.

Input Specification

An amount in the form £X.XX, where X is a digit {0,1,...,9}.

Output Specification

Your program should output the sentence:

```
N coins are needed to make £X.XX
```

where N is the solution, and £X.XX is the amount given in the input. If only one coin is needed you should output

```
1 coin is needed to make £X.XX
```

Sample Inputs.

Input	Output
£0.57	3 coins are needed to make £0.57
£0.50	1 coin is needed to make £0.50

Secret Service Encryption

(If using Java you should name your class 'E')

The Secret Service has introduced a new encryption scheme for integers. Each "secret" integer x is encoded into a sequence of integers such that x is the maximum product of elements in any given contiguous subsequence. Write a decoding algorithm, i.e. an algorithm that, given a sequence of integers, it returns the integer x that the sequence encodes. The integer x is positive, while the integers in the encoding sequence can also be negative. All integers involved are 32-bit, i.e. between $-(2^{31}-1)$ and $2^{31}-1$.

Input Specification

The first line of the input will contain a number n ($1 \leq n \leq 1000000$). This is the length of the encoding sequence. The sequence is described by the following n lines, each of them containing a non-zero integer.

Efficiency notice. Some of the tests are very large. In order to pass all the tests, without running into timeouts or space exhaustion, you will need to write an efficient (linear time) solution.

Output Specification

Your program should output the number encoded by the given sequence.

Sample Inputs.

Input	Output
6 5 6 6 2 -12 12	360
8 5 6 6 2 -12 12 12 -1	622080

Recognising Numbers

(If using Java you should name your class 'F')

You have been asked to write a program that reads off a pixelated image of a number and identifies the number. Each digit (1, 2, 3, 4 and 5) of the given number is represented by a 5 x 7 grid as shown below:

```
...#...#####.#####.#...#...#####.
...#.....#.....#.#...#.#.....
...#...#####.#####.#####.#####.
...#...#.....#.....#.....#.....#
...#...#####.#####.....#...#####.
```

We will only consider numbers containing the digits from 1 to 5.

Input Specification

The pixelated image of a number. The given number could have up to 20 digits. Notice that blocks of 5 x 7 are given without any spaces between them, so, for instance, a number with 5 digits would be given by a 5 x 35 grid.

Output Specification

Your program should output the number shown on the image.

Sample Inputs.

Input	Output
<pre>.#####.#...#...#####.#.#...#.#..... .#####.#####.#####.#####. .#.....#.....#.....#.....# .#####.....#...#####.</pre>	245
<pre>.#####.#...#.#.#...#. .#####.#####. .#.....#.....# .#####.....#.</pre>	24