BHS 2019 Coding Competition

Problem Set

Rules:

1. Each question is worth 50 points. With each incorrect submission, the value of that question decreases by 5 points. You do not need to solve the problems in order.

2. The internet may not be accessed during the contest coding phase. Only one computer per team can be out during this time, meaning phones also need to be put away.

3. The teams with the highest score at the end of the contest will receive awards. In the case of a tie, the team that made their final successful submission first will be the winner.

Problems:

1.Hello!

Input File: None

In *Hello!*, you must print out the statement shown below. There is no input in this problem, but your statement must match **exactly** with the one shown, down to the last dots and spacing.

Input:

None.

Output:

You will output the statement exactly as shown below.

Example Output:

```
I’m excited for the competition! Let's get this bread! \:D/
```

2. Intersect

Input File: intersect.txt

In *Intersect*, you are given two lines and have to determine whether the two lines intersect.

Input:

The first line contains an integer N. The following N lines contains two linear equations separated by spaces. Each equation is the form mx+b, and you may assume that m and b are not zero.

Output:

You must return true if the two lines intersect at at least one point, or false if not.

Example Input:

```
3
2x+6 3x-5
5x+4 5x-1
-3x+5 4x-6
```

Example Output:

```
true
false
true
```

3. Contractions

Input File: contractions.txt

As you finish writing your essay you realize that you’ve been using contractions! Your job is to find and replace all the contractions you used.

Input:

The first line contains an integer N. The following N lines contain sentences with unknown number of contractions.
List of tested contractions:

1. n’t = not
2. ‘ll = will
3. ‘re = are
4. ‘ve = have

You can assume other contractions will not appear.

Output:

Output the sentences with contractions written out as full words.

Example Input:

```
3
We’re almost done
He shouldn't’ve come here
We’ll be there soon
```

Example Output:

```
We are almost done
He should not have come here
We will be there soon
```

4. Tic-Tac-Toe

Input File: tictactoe.txt

You are playing Tic-Tac-Toe in class with your friend. Deciding that it is too hard to determine who won a game, you decide to make a program that does it for you. Your job is to evaluate the given board and return the winner. The winner is determined by having three in a row either horizontally, vertically, or diagonally.

Input:

The first line contains an integer N. There will be N boards to evaluate. A player is represented either as an `X` or an `O`. Each board is a 3 by 3 consisting of space-separated `X`'s and `O`'s. Each board is separated by a space.

Output:

You will output the winner of each board, being either `X`, `O` or Neither.

Example Input:

```
3
X O O
X X O
O O X

O X O
X O X
X O X

O O X
O X X
O X O
```

Example Output:

```
X
Neither
O
```

5. Racers
Input File: racers.txt

You are watching a high school swim event when the computer system broke down. The host asked you to help record the time and show the winners. Using your linux computer, your job is to return the names of the swimmers from fastest to slowest times.

Input:

The first line contains an integer N. The following N lines contain swimmers and their times.

Output:

You will output the a space-separated list of the names of the racers from fastest to slowest.

Example Input:

```
4
Usain 9.81
Phelps 3.14
Woods 2.71
Simpson 6.02
```

Example Output:

```
Woods Phelps Simpson Usain
```

6. Counting Zoo

Input File: zoo.txt

When you got back home from the zoo, your grandmother asks you how many animals were there. You do not remember, so you go back to your photos and try to find out how many animals there were. You look at all your photos and count the total number of animals per species.

Input:

The first line contains an integer N. The following N lines each contain an integer and an animal. The integer represents the number of that animal you saw.

Output:

Output the number and name of the total of each animal in the order that you first saw them.

Example Input:

```
5
5 lions
6 giraffes
3 penguins
4 hippopotamuses
7 penguins
```

Example Output:

```
5 lions 6 giraffes 10 penguins 4 hippopotamuses
```

7. Conversion

Input File: conversion.txt

You are in your science class doing homework problems. You are too lazy to convert the units in each problem, so you write a program to do it for you. Your job is to convert the units from kilometers per hour to meters per second. (1 kilometer = 1000 meters and 1 hour = 3600 seconds).

Input:

The first line contains an integer N. The following N lines contains an integer

Output:

You will output the correct conversion and round down to the nearest integer.

Example Input:

```
4
50
100
250
360
```

Example Output:

```
13
27
69
100
```

8. Significant Figures

Input File: sigfigs.txt

You are now in chemistry. You are doing well, but you cannot quite seem to grasp the concept of significant figures. Significant figures are all numbers except zeros. The only exceptions are any zeros that are surrounded by other numbers, or trailing zeros after the decimal point.

Input:

The first line contains an integer N. The following N lines contain doubles.

Output:

Print the number of significant figures in each double.

Example Input:

```
4
10.1
00450
32.00
01006.0
```

Example Output:

```
3
2
4
5
```

9. StoneHearth

Input File: stonehearth.txt

You are playing a very expensive card game. As your opponent starts taunting you, you want to deal as much damage as possible. However, your opponent has a minion in the way that you need to kill. With your board of minions, your job is to destroy your opponent's minion and deal as much damage.

Input:

The first line contains an integer N. Each of the following N test cases contains the opponent minion's health and your minion's attack separated by spaces.

Output:

Output the highest number

Example Input:

```
3
5
1 3 4 6 8
8
1 4 5 8 12
3
1 8 10
```

Example Output:

```
17
22
11
```

10. Polynomial Division
Input File: division.txt

You are a math teacher and you are making problems for the next test. The test is about polynomial division, and feeling nice, you decide not to give questions that result in remainders. You have already made the problems, but you forgot which have remainders and which don't. Your job is to write a program that determines whether the answer from the input results in a remainder.

Input:

The first line contains an integer N. There are N following equations. Each equations will have two strings, the first being the numerator and the second being the denominator. Note that a term with a coefficient of 1 will have the 1 written, and any constant terms will be multiplied by x^0

Output:

If there is a remainder from the equation, print `true`. If not, print `false`.

Example Input:

```
3
1x^2-9x^1-10x^0 1x^1+1x^0
3x^3-5x^2+10x^1-3x^0 3x^1+1x^0
1x^2+9x^1+14x^0 1x^1+7x^0
```

Example Output:

```
false
true
false
```

11. Missing Number

Input File: missing.txt

You are given a set of numbers it is your job to determine the missing number in each set.

Input:

The first line contains an integer N. The following N lines will have a sequence of 4 space separated integers. These 4 numbers are part of a 5 number set, which increases linearly. You can assume that the missing number will not be on the ends of the set.

Output:

You will output an integer that is missing from each 4 number sequence.

Example Input:

```
5
1 2 3 5
3 5 9 11
101 119 128 137
25 50 100 125
53 84 115 177
```

Example Output:

```
4
7
110
75
146
```

12. Cipher

Input File: cipher.txt

Substitution cipher changes all occurrences of each letter to some other letter. Substitutions shift letters down. For example, in a substitution cypher where letters are shifted down once, A becomes B and B becomes C. Permutation ciphers scramble the order of the letters. For example, applying the permutation ⟨2,1,5,4,3,7,6,10,9,8⟩ to the message “VICTORIOUS” one gets the message “IVOTCIRSUO”.

Input:

The first line contains integer N. The following N lines contain ciphers and possible answers separated by a space. Each cypher is has both a substitution and a permutation cypher applied to it.

Output:

Output `Yes` if the answer a possible result from the cipher, or `No` if it is not.

Example Input:

```
2
JWPUDJSTVP VICTORIOUS
MAMA ROME
```

Example Output:

```
YES
NO
```

14. Boxes

Input File: boxes.txt

It is moving day. You are moving boxes of your stuff into a new house. You have too many boxes to fit into a room, so you want to find the optimal way to store them given the space you have.

Input:

The first line contains an integer N. N represents the number of boxes there are total. The next line contains two integers, the width and length of the room respectively. The following N lines contain the two integers, the width and length of each box.

Output:

You will output the maximum possible number of boxes that can be stored in the room. Boxes can only be used once, but can face in any orientation and be used in any order. Boxes cannot be stacked on top of each other. Not all boxes will be used.

Example Input:

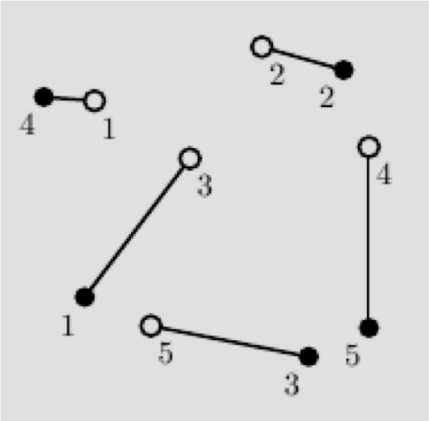
```
5
1 7
1 1
3 2
1 4
2 1
5 4
```

Example Output:

```
3
```

15. Ants
Input File: ants.txt

Young naturalist Bill studies ants in school. Each of the colonies of ants he researches needs food from one apple tree. Bill would like to connect each ant colony to a single apple tree so that none of the paths intersect. Your task is to write a program that find the shortest path between each ant colony and tree. You can assume such a connection is always possible. Below, the picture models the problem. Ant colonies are denoted by empty circles and apple trees are denoted by filled circles. One possible connection is denoted by lines.



Input:

The first line of each dataset contains a single integer number N. N represents the number of ant colonies, as well as the number of apple trees. It is followed by N lines describing N ant colonies, followed by N lines describing N apple trees. Each ant colony and apple tree is described by a pair of integer coordinates x and y on a Cartesian plane. All ant colonies and apple trees occupy distinct points on a plane. No three points are on the same line.

Output:

N lines with one integer per line. Each number represents the shortest path between an ant colony and a tree. The lengths should be given in the order of which the corresponding ants were given.

Example Input:

```
5
-42 58
44 86
7 28
99 34
-13 -59
```

```
-47 -44
86 74
68 -75
-68 60
99 -60
```

Example Output:

```
4
2
1
5
3
```