First Annual University of Central Florida

High School Programming Tournament

Problems

Problem Name	Filename
Roman Numerals	ROMAN
Polynomials	POLY
Pig Latin	PIGLATIN
Contest Scoreboard	CONTEST
Mind Your P's and Q's	PANDQ
Josephus	JOSEPHUS
Monotonous Monotones	MONOTONE
Circular Reasoning	CIRCLE

Call your program file	<Filename>.PAS
Call your input file	<Filename>.IN
Call your output file	<Filename>.OUT

For example, suppose you are solving Polynomials:

Program file:	POLY.PAS
Input file:	POLY.IN
Output file:	POLY.OUT

Roman Numerals

Filename: ROMAN

Roman numerals are an ancient numbering system. If you are unfamiliar with roman numerals, refer to the sheet at the back of the problem set.

The Problem:

Add two roman numerals, expressing the result in roman numerals.

The Input:

Several pairs of roman numerals. Only the capital letters I, V, X, L, C, D and M will be used. Each roman numeral will be on a separate line, starting in column one. The roman numerals and their sums are guaranteed to be within the range of 1 to 3999, inclusive.

The Output:

For each pair of roman numerals, print the sum, expressed in roman numerals. (Capital letters only, please.)

Sample Input:

III
VI
XXIII
XXVII
MCMLXXXIV
DXIV

Sample Output:

IX
L
MMCDXCVIII

Polynomials

Filename: POLY

A *Polynomial* in x is a function consisting of a linear combination of whole number powers of x. The following are polynomials:

$$p(x)=8x^2+2x+1$$

$$q(x)=3x^4-18x^3+13x-44$$

$$r(x)=44x^8-1$$

The highest power of x is called the Order of the polynomial. The order of $p(x)$ is 2, of $q(x)$ is 4, and of $r(x)$ is 8. The sum of two polynomials is also a polynomial. For example:

$$p(x)=x^2+2x+3$$

$$q(x)=4x^2+5$$

$$\begin{aligned} p(x)+q(x) &= (x^2+2x+3)+(4x^2+5) \\ &= (x^2+4x^2)+2x+(3+5) \\ &= 5x^2+2x+8 \end{aligned}$$

The Problem:

Given two polynomials $p(x)$ and $q(x)$, calculate $p(x) + q(x)$.

The Input:

Pairs of polynomials. On the first line, an integer n representing the order of $p(x)$. n is guaranteed to be between zero and ten, inclusive. On the next line is $n+1$ integers, representing the coefficients of $p(x)$, with the coefficient of the nth power of x first, the coefficient of the n-1st power next, and so on, with the coefficient of the 0th power (the constant) last. The information for $q(x)$ follows, in the same format as $p(x)$. There will be several such pairs of polynomials.

The Output:

For each pair of polynomials, print the following

P(X) = *polynomial*
Q(X) = *polynomial*
P(X) + Q(X) = *polynomial*

Print each *polynomial* on a single line, in the format:

a X ^ n + b X ^ (n-1) + ... + w X ^ 2 + y X + z

where a, b, ... z are the coefficients. Print the '+' regardless of whether the next coefficient is positive or negative. **Do not** print the last two terms as X ^ 1 and X ^ 0. If a coefficient is one, don't print it. If a coefficient is zero, omit the term.

Sample Input:

```
2
1 2 3
2
4 0 5
3
48 -12 -10 3
0
5
2
1 1 1
4
1 1 1 1 1
```

Sample Output:

P(X) = X ^ 2 + 2 X + 3
Q(X) = 4 X ^ 2 + 5
P(X) + Q(X) = 5 X ^ 2 + 2 X + 8

P(X) = 48 X ^ 3 + -12 X ^ 2 + -10 X + 3
Q(X) = 5
P(X) + Q(X) = 48 X ^ 3 + -12 X ^ 2 + -10 X + 8

P(X) = X ^ 2 + X + 1
Q(X) = X ^ 4 + X ^ 3 + X ^ 2 + X + 1
P(X) + Q(X) = X ^ 4 + X ^ 3 + 2 X ^ 2 + 2 X + 2

Pig Latin

Filename: PIGLATIN

Pig latin is a simple code. To convert a word to pig latin, break it at the first vowel, put the leading consonants at the end, followed by 'ay.' If the first letter is a vowel, just put 'hay' at the end. For example, 'cat' becomes 'atcay', 'frog' becomes 'ogfray' and 'ant' becomes 'anthay'.

The Problem:

Convert a paragraph into pig latin.

The Input:

A paragraph. Words will either be separated either by a single space, or by a period and a single space. The letter 'Y' will never be the first vowel, so you won't have to handle words like psychology or xylophone. Capital letters will only appear as the first letter of a word. There will be no numbers, contractions, hyphenations, etc. - just words.

The Output:

The paragraph, converted into pig latin. Note that you must maintain proper leading capitalization (Cat becomes Atcay.)

Sample Input:

This is a test of the Pig Latin program. These
simple phrases should just about test all
conditions. Good luck.

Sample Output:

Isthay ishay ahay esttay ofhay ethay Igpay Atinlay ogrampray. Esethay
implesay asesphray ouldshay ustjay abouthay esttay allhay
onditionscay. Oodgay ucklay.

Contest Scoreboard

Filename: CONTEST

Do you remember how this contest is going to be scored? For every problem judged correct, you get one penalty point for every minute from the beginning of the contest until the problem was solved, and twenty penalty points for each judged run. No penalty points are assessed for problems **not** solved. Standings are based on most problems solved, with ties being broken by fewest penalty points.

The Problem:

Write a programming contest scoreboard.

The Input:

The first line has two integers. The first is the number of teams, and the second is the number of problems. On the next line is the data for team 1. The first two numbers are the number of minutes and number of judged runs for the first problem. If the problem was not solved then the number of minutes is given as zero. The next two numbers are the number of minutes and number of judged runs for the second problem, etc. The data for the team 2 follow on the next line, and so on.

The Output:

Print the headings:

PLACE TEAM NUMBER CORRECT PENALTY POINTS

Then, print the appropriate information for each team, in order of finish. Line up the numbers under the appropriate heading so that the last digit of each number is alligned.

Sample Input:

```
 4  3
60  2  120  1    0  3
60  5    0  2  180  1
60  3  120  3  180  3
 0  0    0  1    0  9
```

Sample Output:

PLACE	TEAM	NUMBER CORRECT	PENALTY POINTS
1	3	3	540
2	1	2	240
3	2	2	360
4	4	0	0

Mind Your P's and Q's

Filename: PANDQ

The Problem:

Given a list of strings, count the number of occurrences of the letters **P** and **Q** in each.

The Input:

On each line will be one string, starting in column one. The strings are guaranteed to be between one and seventy characters in length, inclusive. The only characters that will appear in the strings are capital letters.

The Output:

For each input string, print the number of **P**'s and **Q**'s, in the following format:

n P'S, *m* Q'S

where *n* is the number of **P**'s, and *m* is the number of **Q**'s.

Sample Input:

```
GJKRYQRGHBHJBHKOPPIJBHJGUGYUGYIHIHK
PPPPPPPPPQQQQQQQQQQPPPQQQ
THISHASNONEOFEITHER
```

Sample Output:

```
 2   P'S,    1   Q'S
12   P'S,   15   Q'S
 0   P'S,    0   Q'S
```

Josephus

Filename: JOSEPHUS

Josephus is an executioner. To make his job more interesting, he plays the following game: He arranges the condemned prisoners in a circle. Then, he picks a number. Starting from the first man, he counts around the circle until he reaches his number, and executes that prisoner (which, of course, removes that prisoner from the circle.) He starts counting again with the next prisoner in line, and continues until all are dead.

The Problem:

Given the number of prisoners and Josephus' number, determine the order of executions.

The Input:

Each line will have a pair of integers. The first is the number of prisoners, and the second is Josephus' number. Both will be between 1 and 10 inclusive.

The Output:

A list of the prisoners numbers, all on one line, indicating the order in which they died.

Sample Input:

```
4 2
3 4
```

Sample Output:

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

Monotonous Monotones

Filename: MONOTONE

Given a sequence of numbers, a monotonically increasing subsequence is a subsequence in which the numbers are strictly increasing. In this sequence:

1 2 9 4 7 3 11 8 14 6

both of these subsequences are monotonically increasing:

1 2 4 7 11 14 and 1 2 3 8 14

The Problem:

Given a sequence, find the length of the longest monotonically increasing subsequence.

The Input:

Several lines, each with a sequence of integers. There will be no more than ten integers in each sequence.

The Output:

Print each sequence on a single line. On the next line, print the message:

THE LENGTH OF THE LONGEST INCREASING SUBSEQUENCE IS n

Then, print a blank line.

Sample Input:

```
1 2 9 4 7 3 11 8 14 6
1 2 3 4 5
5 4 3 2 1
```

Sample Output:

```
  1  2  9  4  7  3 11  8 14  6
THE LENGTH OF THE LONGEST INCREASING SUBSEQUENCE IS  6

  1  2  3  4  5
THE LENGTH OF THE LONGEST INCREASING SUBSEQUENCE IS  5

  5  4  3  2  1
THE LENGTH OF THE LONGEST INCREASING SUBSEQUENCE IS  1
```

Circular Reasoning

Filename: CIRCLE

The Problem:

Given three non-collinear points, determine the center and radius of the unique circle passing through all three.

The Input:

Several lines, each with six real numbers, in the order

x1 y1 x2 y2 x3 y3

Where (x1,y1), (x2,y2) and (x2,y3) are the three non-collinear points.

The Output:

For each set of points, print out the center and radius of the requested circle, clearly labelled, and accurate to two decimal places.

Sample Input:

```
-1.00  0.00  1.00  0.00  0.00  1.00
 0.00  0.00  0.50  0.50  1.00  0.00
```

Sample Output:

```
Center: (  0.00,  0.00)  Radius:  1.00
Center: (  0.00,  0.50)  Radius:  0.50
```

Roman Numeral Reference

Roman Numeral Symbols:

I	1
V	5
X	10
L	50
C	100
D	500
M	1000

The roman numeral representation of a number is the shortest possible combination of the above symbols which add up to the number, arranged in descending order. 4's and 9's are handled specially, by subtraction. A smaller symbol is placed befor a larger one, and is subtracted from it. For example, IV is four, and XC is ninety. The representation of each number is **unique**- there can only be **one** representation for any number.

I	1	X	10	C	100	M	1000
II	2	XX	20	CC	200	MM	2000
III	3	XXX	30	CCC	300	MMM	3000
IV	4	XL	40	CD	400		
V	5	L	50	D	500		
VI	6	LX	60	DC	600		
VII	7	LXX	70	DCC	700		
VII	8	LXXX	80	DCCC	800		
IX	9	XC	90	CM	900		

For example, 1987 is MCMLXXXVII