University Interscholastic League

Computer Science Competition

2004 Regional Programming Set

Judges' Answers

I. General Notes

1. Unless the exact formatting is specifically part of the problem, an answer should NOT be judged wrong for minor formatting variations such as indent/no indent, extra/no blank lines, and so forth.

2. The answer is only correct if their program successfully runs ALL of the judge's data sets for a given problem.

3. Note that the input data file for each problem begins with the examples from the problems (or a slight variation), and then goes on to more complex cases. The testing is (by definition) not exhaustive in any sense and it is of course possible that an incorrect program will pass all of the tests provided.

II. Point Values and Names of Problems

Number	Name	Point Value
Problem 1	This Is Correct!	6
Problem 2	Test Scores	6
Problem 3	List the Primes	6
Problem 4	Arena	6
Problem 5	Dice Golf	6
Problem 6	Matrix	6
Problem 7	Matrix Reloaded	6
Problem 8	PacMan	6
Problem 9	drawkcaBsay iPgay itaLnay Day	6
Problem 10	Roman Numeral Translator	6
Total		**60**

Problem 0	Dry Run	0 Points

Program Name: dryrun.java **Input File: dryrun.dat**

Write a program that reads a list of items from the input file and outputs a message for each.

Input
The first line contains an integer, *n*, that indicates how many items are in the input file. The next *n* lines contain a single word. Each word represents an item that you like.

Output
For each item in the input, output a line stating, "`I like <item>.`". For example, if the item were cabbage, the program would output the line, "`I like cabbage.`"

Example Input File
```
4
cabbage
contests
judges
everything
```
Example Output To Screen
```
I like cabbage.
I like contests.
I like judges.
I like everything.
```

Problem 1

This Is Correct!

6 Points

Program Name: correct.java **Input File: [none]**

Input File

```
[no input]
```

Output to screen

```
This Is Correct!
```

Problem 2

Test Scores

6 Points

Program Name: scores.java **Input File: scores.dat**

Input File

```
0 James
100 Laura
99 Tim
98 Marc
50 Buddy
```

Output to screen

```
James 0
Laura 100
Tim 99
Marc 98
Buddy 50
```

Problem 3	List the Primes	6 Points

Program Name: primes.java **Input File: primes.dat**

Input file

```
2 10
20 20
100 120
999 999
2 100
```

Output to screen

```
2 3 5 7
No primes found!
101 103 107 109 113
No primes found!
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97
```

Problem 4	Arena	6 Points

Program Name: arena.java **Input File: arena.dat**

Input File

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

Output to screen

```
Fighter 2 is the victor!
Fighter 2 is the victor!
This battle ended in a draw!
This battle ended in a draw!
Fighter 1 is the victor!
```

Problem 5

Dice Golf

6 Points

Program Name: golf.java Input File: golf.dat

Input File

```
20
4 4 3 3 4 4 3 3 4 4 3 3 4 4 3 3 4 4 3 2
2
2 2
3
3 5 2
15
6 3 8 5 12 4 4 4 4 5 11 4 5 2 7
9
4 3 4 3 4 3 4 3 2
18
4 5 9 10 6 8 11 12 4 5 4 5 4 5 3 7 5 2
17
6 4 6 4 6 4 6 4 6 7 6 11 6 12 6 5 6
5
7 7 2 7 7
```

Output to screen

```
Player 2 wins!
Player 1 wins!
Player 1 wins!
Player 2 wins!
Player 1 wins!
Player 2 wins!
Player 2 wins!
Player 1 wins!
```

Problem 6

Matrix

6 Points

Program Name: matrix.java Input File: matrix.dat

Input File

```
7
3
8 1 6
3 5 7
4 9 2
4
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
3
9 2 7
4 6 8
5 10 3
4
16 3 2 13
5 10 11 8
9 6 7 12
4 15 14 1
10
1 92 3 94 5 6 97 98 99 10
20 89 18 87 16 15 84 13 82 81
21 72 23 74 25 76 77 28 79 30
40 69 38 67 35 36 64 33 62 61
41 52 43 44 56 55 57 48 59 50
51 42 58 54 46 45 47 53 49 60
70 39 68 37 66 65 34 63 32 31
80 22 73 24 75 26 27 78 29 71
90 19 88 17 86 85 14 83 12 11
91 9 93 7 95 96 4 8 2 100
1
1
2
1 2
3 4
```

Output to screen

```
This magic square has sum = 15.
This isn't a magic square.
This isn't a magic square.
This magic square has sum = 34.
This magic square has sum = 505.
This magic square has sum = 1.
This isn't a magic square.
```

Problem 7

Matrix Reloaded

6 Points

Program Name: reloaded.java **Input File: reloaded.dat**

Input File

```
13
3
8 1 6
3 0 7
4 9 2
2
0 2
3 0
2
1 0
0 0
5
9 9 9 9 9
0 9 9 9 0
9 9 9 9 9
0 9 9 9 9
9 9 9 9 9
10
0 3 2 9 8 3 5 1 8 8
3 2 1 6 4 2 1 4 6 3
2 1 2 2 2 1 1 5 3 2
9 8 2 2 1 3 4 2 1 9
8 8 9 3 1 3 1 1 3 8
3 2 1 4 3 0 1 3 3 3
5 4 3 2 9 4 3 2 4 5
1 3 5 9 2 1 3 4 2 1
8 9 2 1 2 4 5 3 2 8
3 3 2 9 8 3 5 1 8 0
3
8 1 6
3 0 7
4 9 2
4
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
3
9 2 7
4 6 8
5 10 3
4
16 3 2 13
5 10 0 8
9 6 7 12
4 15 0 1
10
1 92 3 94 5 6 97 98 99 10
20 89 18 87 16 15 84 13 82 81
21 72 23 74 25 76 77 28 79 30
40 69 38 67 35 36 64 33 62 61
41 52 0 44 56 55 57 48 59 50
51 42 58 54 46 45 47 53 49 60
70 39 68 37 66 65 34 63 32 31
```

```
80 22 73 0 75 26 27 78 29 71
90 19 88 17 86 0 14 83 12 11
91 9 93 7 95 96 4 8 2 100
1
1
2
1 2
3 4
3
8 0 0
3 0 7
4 9 2
```

Output to screen

```
This could be a magic square.
This can't be a magic square.
This can't be a magic square.
This can't be a magic square.
This can't be a magic square.
This could be a magic square.
This can't be a magic square.
This can't be a magic square.
This could be a magic square.
This could be a magic square.
This could be a magic square.
This can't be a magic square.
This could be a magic square.
```

Problem 8

PacMan

6 Points

Program Name: pacman.java **Input File: pacman.dat**

Input File

```
12
5
#####
#C#X#
#.#.#
#.A.#
#####
7
#######
#.....#
#.###.#
#..X#.#
#####A#
#@...C#
#######
4
####
#C##
##X#
####
4
####
#CA#
#AX#
####
4
####
#CA#
#.X#
####
7
#######
#@....#
#.ACA@#
#.#A#.#
#...X.#
#######
#######
7
#######
#.....#
#.#C#@#
#@#A#.#
#.#####
#....X#
#######
10
##########
#CAX.....#
#.######.#
#.#....#.#
#.#.##.#.#
#.#.##.#.#
#.#.#@.#.#
```

```
#.#.##.#.#
#........#
##########
10
##########
#CAX.....#
#.######.#
#.#....#.#
#.#.##.#.#
#.#.##.#.#
#.#.#..#.#
#.#@##.#.#
#........#
##########
10
##########
#@.......#
#........#
#........#
#........#
#........#
#........#
#......CA#
#......AX#
##########
10
##########
#@A......#
#A.......#
#........#
#........#
#........#
#........#
#......CA#
#......AX#
##########
10
##########
#CX......#
#......@.#
#........#
#..@.A...#
#........#
#...A.@..#
#........#
#.A..@.A.#
##########
```

Output to screen

```
PacMan should retire.
PacMan can escape in 20 moves.
PacMan should retire.
PacMan should retire.
PacMan can escape in 2 moves.
PacMan can escape in 7 moves.
PacMan can escape in 11 moves.
PacMan can escape in 26 moves.
PacMan can escape in 22 moves.
PacMan can escape in 26 moves.
PacMan should retire.
PacMan can escape in 1 moves.
```

Problem 9

drawkcaBsay iPgay itaLnay Day

6 Points

Program Name: day.java Input File: day.dat

Input File

```
Hello nice to meet you
This is a funny way to talk
We like to eat pork and fish
We like green beans
We are cats
Hello world
Hi
I am a problem
A problem am I
```

Output to screen

```
lleHoay cineay toay eemtay oyuay
ihTsay isay aay nnufyay awyay toay latkay
Weay kileay toay aetay ropkay naday sifhay
Weay kileay eergnay naebsay
Weay raeay tacsay
lleHoay lrowday
Hiay
Iay amay aay elborpmay
Aay elborpmay amay Iay
```

Problem 10

Roman Numeral Translator

6 Points

Program Name: roman.java **Input File: roman.dat**

Input File

```
III
XIV
CCLXXXIX
I
IV
MMDCCCLXXXVIII
CMXLIV
CMXXIX
MDCLXVI
MMIM
MMCMIC
MMCMXCIX
C
CMXCIX
```

Output to screen

```
3
14
289
1
4
2888
944
929
1666
2999
2999
2999
100
999
```