University Interscholastic League

Computer Science Competition

2004 State 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.

4. All problems have a value of 6 points.

II. Point Values and Names of Problems

Number	**Name**	**Point Value**
Problem 1	Average Joe	6
Problem 2	Bowling for Dummies	6
Problem 3	Function Finder Fun	6
Problem 4	Radiant Primes	6
Problem 5	Scheduling Nightmare on Elm Street	6
Problem 6	Bust ‘A Move	6
Problem 7	Poly’s Nomials	6
Problem 8	Who’s the Boss?	6
Problem 9	Enemy at the Gates	6
Problem 10	Let’s Make a Meal	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: Average Joe

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

Input File

```
5
3
James 100
Marc 50
Tim 75
7
Tim 75
Marc 50
James 100
Marc 110
Tim 0
Sparky 5
James 0
5
Marc 100
Marc 50
Marc 79
Marc 90
Marc 100
10
Marc 100
Marcie 40
Marcell 77
Mac 50
Mac 100
Marcell 90
Marcie 100
Marc 74
Marvin 60
Marcie 64
8
Jimbob 93
Joebob 92
Joe 83
Jim 82
Janebob 73
Jane 72
Janedoe 60
Johndoe 59
```

Output to screen

```
3 students
James A
Marc F
Tim C
4 students
James F
Marc C
Sparky F
Tim F
1 students
Marc B
5 students
Mac C
Marc B
Marcell B
Marcie D
```

```
Marvin D
8 students
Jane D
Janebob C
Janedoe D
Jim C
Jimbob A
Joe B
Joebob B
Johndoe F
```

Problem #2: Bowling for Dummies

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

Input File

```
8
8 / 9 / X X 3 6 7 / 8 1 4 / X X 6 /
8 1 9 0 X X 3 6 7 2 8 1 4 5 9 / 3 / 2
X X X X 9 / X 8 1 X 7 / X X X
X X X X X X X X X X X X
X X X X X X X X X 9 / X
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 / X 0 0 1 / 5 0 X 0 / 2 3 4 5 6 3
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 / X 5 /
```

Output to screen

```
183
121
227
300
279
0
105
40
```

Problem #3: Function Finder Fun

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

Input file

```
int x,y;
int z=(3 + 2) * 5;

System.out.println("Test");

for(x = 0; x < 20; x++)
   for(y = 0; y < 30; y++)
      if(y == 1)
         System.out.println(x + "+" + y + "=" + (x + y));

while(z <= 25)
{
   foo(z);
   parse(z + 3);
   z++;
}

switch(z) {
    case 0:  valueswitch(z);
             break;
    default: returnerror(getreturnError());
}

if(getString().equals("yes"))
{
    System.out.println("The results is " + getString());
} else {
    System.out.println("Could not find result");
}

urn() or() witch();

return(x * y * z);

/*
 * This is the end
 * of this program
 */
```

Output to screen

```
println
foo
parse
valueswitch
returnerror
getreturnError
getString
equals
urn
or
witch
```

Problem #4: Radiant Primes

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

Input File

```
8
11 2
17 3
2 9
2 2
1000 2
1000 9
13 5
23 7
```

Output to screen

```
prime
non-prime
prime
non-prime
non-prime
non-prime
prime
prime
```

Problem #5: Scheduing Nightmare on Elm Street

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

Input File

```
5 13
Enterprise
CSI Miami
CSI
Survivor
The Practice
2130 CSI
0130 The Practice
1900 Enterprise
1900 CSI
2000 CSI
2000 Survivor
1900 The Practice
2000 CSI Miami
2130 The Practice
2200 CSI Miami
0130 Survivor
2000 The Practice
2000 Enterprise
```

Output to screen

```
Survivor
Enterprise
CSI Miami
CSI
```

Problem #6: Bust 'A Move

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

Input File

```
7
..........
.RR.......
.RR.......
..........
..........
..........
.R.R.BBB..
.RRR..B...
.R.R.BBB..
..........
RRRRRRRRRR
BBBBBBBBBR
RRRRRRRRRR
RGGGGGGGGG
RRRRRRRRRR
RRRRRRRRRR
R........R
R........R
R........R
RRRRRRRRRR
RRRRRRRRRR
RYYYYYYYYR
RYRRRRRRYR
RYYYYYYYYR
RRRRRRRRRR
..........
..........
..........
..........
..........
YYRRGGBBYY
RRGGBBYYRR
GGBBYYRRGG
BBYYRRGGBB
YYRRGGBBYY
```

Output to screen

```
1 groups
2 groups
3 groups
1 groups
3 groups
0 groups
25 groups
```

Problem #7: Poly's Nomials

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

Input File

```
7
7 + 4x^8 - 34x^2 - x
x^8
- 17 + x^8 - 9x^7 - 243x^9 - x^5
- 1 - 2x - x^4 - 100x^3
- 1
- x
- x^9 + 1000x - 1000x^8 + 100x^2 - 100x^7 + 10x^3 - 10x^6 + 50x^4 - 50x^5
```

Output to screen

```
4x^8 - 34x^2 - x + 7
x^8
- 243x^9 + x^8 - 9x^7 - x^5 - 17
- x^4 - 100x^3 - 2x - 1
- 1
- x
- x^9 - 1000x^8 - 100x^7 - 10x^6 - 50x^5 + 50x^4 + 10x^3 + 100x^2 + 1000x
```

Problem #8: Who's the Boss?

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

Input File

```
6
3 Burns Homer Lenny Carl
2 Lovejoy Ned Maude
2 Homer Bart Lisa
1 Marge Maggie
2 Ned Todd Rod
1 Bart SantasLittleHelper
6
Bart Todd
Burns Homer
Todd Lovejoy
Lenny Homer
SantasLittleHelper Burns
Maude Lovejoy
```

Output to screen

```
Bart: No Todd, I don't have time to do your work and mine.
Burns: No Homer, I don't have time to do your work and mine.
Todd: Sure Lovejoy, I'll get right on it.
Lenny: No Homer, I don't have time to do your work and mine.
SantasLittleHelper: Sure Burns, I'll get right on it.
Maude: Sure Lovejoy, I'll get right on it.
```

Problem #9: Enemy at the Gates

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

Input File

```
11
2
1 1
A
8
1 0 0 1 1 1 0 0
A A R R X A X
2
1 1
A
2
1 1
R
2
1 1
X
2
1 0
A
2
0 1
R
2
1 0
X
8
1 1 1 1 0 0 0 0
A A A A A A R
32
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
A X R A X R A X R A X R A X R A X R A X R A X R A X R A X R A
8
1 0 1 0 0 0 0 0
R R R R A A X
```

Output to screen

```
1
0
1
1
0
0
1
1
1
0
1
```

Problem #10: Let's Make a Meal

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

Input File

```
8
4
Carrots
Orange Soda
Chocolate Cake
Hamburger
6
Red Soda
Coffee
Chocolate Candy
Potato Chips
Cheeseburger
Buttered Popcorn
4
Coffee Cake
Coffee Cake
Pizza Chips
Pizza Chips
8
Yummycoffee
Yummysoda
Yummycake
Yummycandy
Yummychips
Yummypopcorn
Yummyburger
Yummypizza
5
Brocolli
Carrots
Steak
Water
Bread roll
4
Carrot Candy
Tuna Pizza
Water
Coffee Candy Pizza Chips
4
Baking soda
Poker Chips
Pizza Hut Coupon
Urinal cake
4
Coffee Candy Pizza Chips
Coffee Candy Pizza Chips
Coffee Candy Pizza Chips
Coffee Candy Pizza Chips
```

Output to screen

```
What would your mom say?
Programmer Fuel
Programmer Fuel
Programmer Fuel
What would your mom say?
Programmer Fuel
Programmer Fuel
What would your mom say?
```