A+ Computer Science October 2013

Computer Science Competition

Hands-On Programming Set

I. General Notes

1. Do the problems in any order you like. They do not have to be done in order from 1 to 12.
2. All problems have a value of 60 points.
3. There is no extraneous input. All input is exactly as specified in the problem. Unless specified by the problem, integer inputs will not have leading zeros. Unless otherwise specified, your program should read to the end of file.
4. Your program should not print extraneous output. Follow the form exactly as given in the problem.
5. A penalty of 5 points will be assessed each time that an incorrect solution is submitted. This penalty will only be assessed if a solution is ultimately judged as correct.

Number	Name
Problem 1	Mean
Problem 2	Twitter
Problem 3	Snake
Problem 4	Similar
Problem 5	Refraction
Problem 6	DoubleT
Problem 7	CSMeals
Problem 8	Climb
Problem 9	Sexy
Problem 10	Synonyms
Problem 11	Triple
Problem 12	Checkers

Good luck!

1. Mean

Program Name: Mean.java Input File: mean.dat

One common math lesson in elementary school is the mean, median, mode. The mean is the average. The median is the middle number (or average of the two middle numbers in a data set with even number of elements). The mode is the most common occurring element. In this program, there will be only one mode in a set of numbers.

Your task is the find the mean, median and mode of a list of integers. Each list will be on one line of at least 3 numbers and no more than 20 positive integers. The largest number will be 99 (2-digit numbers).

Input: The first line consists of the number of data sets. Each subsequent line contains from 3 to 20 integers of random order.

Output: The first line of output will be `"mean median mode"`. Each subsequent line will have the mean, median, and mode printed out below the words aligned right-justified. Print out all numbers as decimals to the nearest tenth.

Example Input File

```
3
1 2 3 3
10 10 10 10 10 10 10 10 10 10
2 3 4 5 5 9 8 7 5 5 3 2 1
```

Output to screen:

```
mean median mode
 2.3    2.5  3.0
10.0   10.0 10.0
 4.5    5.0  5.0
```

2. Twitter

Program Name: Twitter.java   Input File: twitter.dat

A tweet can have at most 140 characters. I saw an article somewhere that said the average tweet was 67.9 characters and the median was 60. Write a program that determines if a short message can be a tweet!

For this program, tweets will start with the "@" symbol and have 140 characters or less (counting spaces between words). There will be no space at the end of a line.

Input: The first line consists of the number of data tweets in the file. Each tweet starts with the "@" symbol. There could be at most 3 lines per tweet. No line will end with a space.

Output: Print either "tweet" or "not tweet" for each message.

Example Input File
```
3
@coolstuff This tweet will have exactly 81 characters
so it is definitely a tweet!
@funnyman This message is really, really, really long so it will
not be a tweet.  No one wants a long tweet cuz ain't nobody got time for
that!
@sweetbrown I am so awesome that Beyonce quotes me!
```

Output to screen:
```
tweet
not tweet
tweet
```

3. Snake

Program Name: Snake.java Input File: snake.dat

Many people have played the snake game (a few years ago Nokia phones came with snake on it, sometimes when watching YouTube videos you could pause the video and press the left arrow to start the game…).

Let's simulate that game!

For this program, you will be given a matrix (15x15) of spaces and letters. The snake will consist of 3 contiguous 'X' characters, and the food pellets are represented by the 'F' character. The object of the game is to eat food pellets and grow, without hitting the boundary (stay within the size of the matrix). However, for this program, the snake will not grow, nor can it collide with itself. Just try to eat the food pellet and not exit the matrix. (No out of bounds exceptions!) You only have to consider the matrix cell that contains the head of the snake (ignore the other two cells for this program).

The input will be a string of 20 of the characters "UDLRO" standing for Up, Down, Left, Right and Onward…continue in the same direction. For each input, determine either how many food pellets are eaten, or if it is game over (by leaving the matrix). The game will restart with the same board configuration of snake and pellets. The snake will start to move to the RIGHT at the beginning of the game.

Here is an example of the snake moving with the following input string:

```
UROOOUOOLOOOOOUOOOOO
12345678901234567890
```

```
  F     F
        7
        6
    12345
  XXX    F
```

On the next move (move 8), the snake would eat the pellet! Then, you would turn left and continue to eat the next pellet straight ahead (move 14).

Input: The first 15 lines consists of the matrix (15x15 characters). The next line will consist of the number of data sets. Each data set will consist of a line of 20 characters each (UDLRO).

Output: For each data set, print out "N pellets" or "GAME OVER".

(Continued on next page…)

(Problem 3 contin.)

Example Input File
```
F             F
        F    F
 F
 F

  F     F



  XXX    F


      F  F

              F
2
UROOOUOOLOOOOOOUOOOO
UOOOOOOOOOOOOOOOOOOO
```

Output to screen:
```
4 pellets
GAME OVER
```

4. Similar

Program Name: Similar.java Input File: similar.dat

Similar triangles have equal corresponding angles, and the corresponding sides are in proportion to each other. Given the three sides of one triangle and two sides of a similar triangle, find the 3rd side. All sides will be integers (no decimals).

Input: The first line consists of the number of data sets in the file. Each subsequent line will have 5 integers, the 3 sides of one triangle and the 2 sides of the similar triangle.

Output: For each data set, print out the three sides of the missing similar triangle.

Example Input File
```
3
3 4 5 6 8
10 12 10 5 6
2 3 4 6 9
```

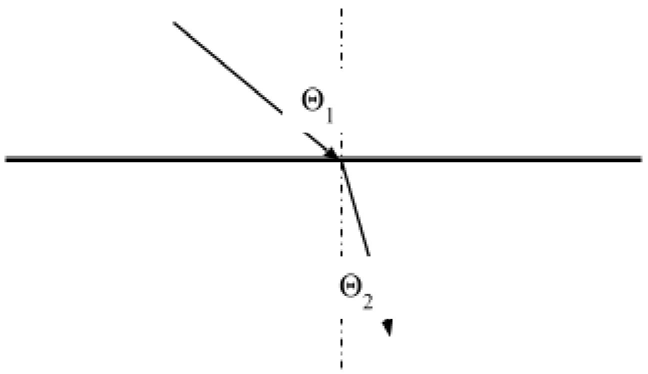
Output to screen:
```
6 8 10
5 6 5
6 9 12
```

5. Refraction

Program Name: Refraction.java   Input File: refraction.dat

Snell's Law in physics governs the angles of incident and refracted light. Light travels with a different velocity through different transparent materials. This is why a straw looks bent in a glass of water, and why when you step in water it is deeper than it appears.

Use Snell's Law to calculate either an angle of refraction or the index of refraction.



The angle (θ) in degrees is measured to the normal, which is perpendicular to the boundary of the 2 materials. The index of refraction (n) is a decimal 1.000 or higher, and the angles can be from 0^o to 90^o. The angle and index of refraction are inversely related (when n is large, the angle is small) according to the following equation:

$$n_1 \sin sin\, \theta_1 = n_2 \ \sin sin\, \theta_2$$

HINT – You might need to look up the Math.sin() method in the Java documentation.

Input: The first line consists of the number of data sets in the file. Each data set is on one line. The index and angle is given for the first object, and either the index or angle is missing for the second material.

Output: For each data set, print out either "n = ____ " or " angle = ____". Round n to the nearest thousandth and the angle to the nearest degree.

Example Input file

```
3
1.000 30 1.333 angle
1.25 50 n 40
2.419 24 1.000 angle
```

Output to screen:

```
angle = 22
n = 1.490
angle = 80
```

6. DoubleT

Program Name: DoubleT.java   Input File: no data file

Because I graduated from Texas Tech in 1997, this program is dedicated to the Red Raiders. No matter what you think about Lubbock, you've got to admit, Texas Tech has had an effect on Big 12 athletics in the past. I remember being a freshman when Sheryl Swoopes and Marsha Sharp won a national title in women's basketball! I can recall quite a few good football games that were high scoring with exciting players. Here's a few I remember: Zach Thomas, Bam Morris, Wes Welker, Kliff Kinsgbury, Graham Harrell, Michael Crabtree, Seth Doege…

If you like Tech, think of all the awesome victories!

If you don't like Tech, well, you can remember some other times as well.

Just draw the Texas Tech double T logo. Make it 10 characters tall, 15 across, using the capital letter T.

GO RED RAIDERS!

Input: No input file.

Output: Print out the picture below.

Output to screen:

```
TTTTTTTTTTTTTTT
T     TTT     T
T     TTT     T
      TTT
   TTTTTTTTT
   T  TTT  T
      TTT
      TTT
      TTT
      TTT
```

7. CSMeals

Program Name: CSMeals.java Input File: csmeals.dat

Dave Ramsey is a financial counselor who teaches “Debt is dumb, cash is king.” To stay on a tight budget, a good way to watch your spending is to have a food budget and not eat out. Food preparation takes more time, but you can eat for less money. Sometimes, when you go to the grocery store and you’re hungry, you tend to spend more money on impulse purchases. So make a list and follow it. Write this program to guide you to financial peace!

You will be given a grocery list. Then, you will be given the receipt for the purchases. Decide whether the list is one of three possibilities:

1. OK (purchases are equal to list)
2. you need to get more items on your list (you forgot items)
3. you have impulse purchases and you are not on the plan (but you did purchase everything else on the list)

Both lists will be in random order.

Input: The first line will contain the number of data sets. Each data set consists of a grocery list and a receipt (list of purchases). The first line of each data set is the size of each list, N. There will be N lines (no more than 5) of each list. Each line will contain less than 10 one-word items. The grocery list will be the first N lines and the receipt will be the last N lines in each data set. All items on the list will be lowercase.

Output: There will be three possibilities for output:

1. Print “OK” is the grocery list and receipt match exactly.
2. Print “BUY MORE _____” indicating what you need to buy to complete the list.
3. Print “PUT BACK ______” indicating what items are on the receipt but not on the list.

Separate items with a space if there are more than 1, in the order they appear in the list or receipt.

(Continued on next page…)

Example Input file

```
3
1
eggs ham bread
ham eggs bread
2
eggs ham bread lettuce poptarts flour sugar tea hamburger
tortillas beans corn tomatoes
toy eggs ham bread lettuce poptarts flour sugar tea
hamburger candybars corn tomatoes beans tortillas icecream
3
lettuce carrots tomatoes celery cabbage beans tortillas
pork beef chicken ribs bread crackers chips cookies milk
oj caprisun drpepper
lettuce tomatoes celery cabbage beans tortillas
pork beef chicken bread crackers cookies milk
caprisun drpepper
```

Output to screen:

```
OK
PUT BACK toy candybars icecream
BUY MORE carrots ribs chips oj
```

8. Climb

Program Name: Climb.java Input File: climb.dat

Given a number, draw a triangle of that size with the character 'C.'

Input: The first line consists of the number of data sets in the file. Each data set consists of a single integer greater than 1 and less than 30.

Output: Print out the triangle for each data set, separated by a blank line.

Example Input file

```
3
2
5
8
```

Output to screen:

```
 C
CC

    C
   CC
  CCC
 CCCC
CCCCC

       C
      CC
     CCC
    CCCC
   CCCCC
  CCCCCC
 CCCCCCC
CCCCCCCC
```

9. Sexy

Program Name: Sexy.java Input File: sexy.dat

Did the program title get your attention? Well, it is another boring old program about prime numbers ☹.

Sexy primes are two prime numbers that have a difference of 6 (sext- is the Latin prefix for six). Given two numbers, determine if they are both prime and if they have a difference of 6. For example, the numbers 5 and 11 are both prime and the difference is 6. They are sexy primes!

For this program, the largest value will be 30,000.

Input: There will be an unknown number of data sets. Each data set will consist of two integers, smaller number first.

Output: Print out either "SEXY" or "NOT" for each data set.

Example Input file

```
5 11
7 13
2 8
433 443
641 647
```

Output to screen:

```
SEXY
SEXY
NOT
NOT
SEXY
```

10. Synonyms

Program Name: Synonyms.java Input File: synonyms.dat

In English class, you learn that writing is an art form. You can use words like paint on a canvas! Using the right words can make an average paper turn awesome. But some teenagers do not have a love for the written word—they just want to get it done! You will write a program to automate this process.

Given a paragraph filled with the same word twice or more, you will fill in a synonym for the word to add variety. Some words may appear more than twice, so you will be given a list of possible synonyms. The first word will have several synonyms following it to be replaced with, but not all must be used. Keep the first occurrence of the word in the paragraph. Replace each successive word with the next synonym in the order given.

To keep this program simpler:

- punctuation will be removed
- all words will be lowercase
- all words will be separated by a single space

Input: The first line (X) will indicate the number of lines in the paragraph. The next X lines contain the paragraph to be edited. The next line (Y) will indicate how many synonyms are in the data set. The next Y lines contain a word followed by a list of synonyms to be used.

Output: Print out the modified paragraph.

Example Input file

```
4
this paragraph was fun to write i love to write fun assignments
having work to do is very fun my favorite subject is english but
i also love to write programs which is a close second but not bad this bad
assignment was bad bad bad bad and bad
4
fun diverting festive
love enjoy relish
write scribe code
bad horrible bitter terrible evil base rancid
```

Output to screen:

```
this paragraph was fun to write i love to scribe diverting assignments
having work to do is very festive my favorite subject is english but
i also enjoy to code programs which is a close second but not
bad this horrible assignment was bitter terrible evil base and rancid
```

11. Triple

Program Name: Triple.java Input File: triple.dat

You have three numbers. Print out those numbers tripled. That's it!

Input: There are three lines in the data file, each containing one integer

Output: Show the three numbers tripled.

Example Input file

```
1
2
3
```

Output to screen:

```
3
6
9
```

12. Checkers

Program Name: Checkers.java Input File: checkers.dat

For fun family time, my daughters and I play checkers. First, I am teaching them the rules. Then we learn some strategy and thinking skills. I let them win more than I do to keep it fun, but I show them how they could do better in the next game.

In the game of checkers, you can jump over the opponent checkers diagonally if the next diagonal space is empty. You can also do multiple jumps going forward. Let's write a program to determine which move you should make — which move would have the highest number of jumps!

You will be given a checkerboard in the middle of a game. You will be the red player (at the bottom of the board) playing against black (at the top). Find the position of the red checker (row and column in the matrix) that would have the maximum number of jumps. In this program, there will be one clear maximum (no ties). There will always be at least one jump. The maximum for the board is three jumps.

For example, on the following checkerboard, the R at row 5 and column 3 can jump twice, while the R checker at [5,1] or [4,4] can only jump 1 time.

				B		B	
	B				B		B
B		B		B			
					B		B
B		B		R			
	R		**R**		R		R
R		R		R		R	
	R		R		R		

```
    B B
 B   B B
B B B
     B B
B B R
 R R R R
R R R R
 R R R
```

Input: The first line contains the number of data set (checkerboards). Each checkerboard has 8 lines of 8 characters each (one of three letters, R, B or space)

Output: For each board, print out the position of the red checker and the maximum number of jumps.

(Continued on next page…)

(Problem 12 contin.)

Example Input file

```
2
    B B
 B   B B
B B B
     B B
B B R
 R R R R
R R R R
 R R R
B B B B
     B
B B B R
       R
  B
 R     R
R B R
 R R R R
```

Output to screen:

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