PSCSTA Programming Contest
Dec 2015 - Novice Division

DO NOT OPEN THIS PACKET UNTIL INSTRUCTED TO DO SO

General Notes

1. Do the problems in any order you like.
2. All problems have a value of 60 points. Incorrect submissions may be reworked and resubmitted, but will receive a deduction of 5 points for each incorrect submission. Deductions are only included in the team score for problems that are ultimately solved correctly.
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. Your program should work for all expected input cases, as specified in the problem. Sample test cases may be simple. Judge test cases could be more involved.

#	**Name**
1	JarJarAwakens
2	Times Tables
3	Gift1
4	Gift2
5	Read1
6	Read2
7	DandD1
8	Rain
9	Add
10	Fives
11	Pay in Cash
12	Flipping Lights

Good luck!

1. JarJarAwakens

Input File: none

J. J. Abrams (the director of Star Wars Episode 7) has been taken over by a Sith Lord! He has decided that the new name of Episode 7 will be Jar Jar Awakens! In this new story line, Jar Jar Binks had been frozen in carbonite and stored by the Emperor and Darth Vader in Episode 6 before the 2nd Death Star was destroyed. When the Sith return, they will have the secret weapon of Jar Jar Binks to destroy the galaxy and bring a new reign of terror.

In this program, print out the word "no" seven times, lowercase, no punctuation, 1 space between each word.

Input: none

Output: Print out the word "no" seven times, lower case, no punctuation, 1 space between each word.

Example Input file: none

Example Output to Screen

```
no no no no no no no
```

2. Times Table

Input File: tables.dat

Don't like to memorize the times tables? Let us write a program to generate it whenever we need it!

Input
The first line will contain an integer N, which represents the number of requests below. Subsequent N lines have the integers Ti whose tables should be generated and output.

Constraints

1<=N<=10
1<=Ti<=1000

Output
Output the requested tables from times 1 to times 10, for each requested table. There should be N such tables, each separated by a blank line.

Example Input File

```
2
2
5
```

Output to Screen

```
2 X 1 = 2
2 X 2 = 4
2 X 3 = 6
2 X 4 = 8
2 X 5 = 10
2 X 6 = 12
2 X 7 = 14
2 X 8 = 16
2 X 9 = 18
2 X 10 = 20

5 X 1 = 5
5 X 2 = 10
5 X 3 = 15
5 X 4 = 20
5 X 5 = 25
5 X 6 = 30
5 X 7 = 35
5 X 8 = 40
5 X 9 = 45
5 X 10 = 50
```

3. Gift1

Input File: gift1.dat

You get a gift card for your birthday. Each gift card comes pre-loaded with a certain balance, and there is list of items you want to buy, listed in order of your favorite items. Determine how much of the list you can buy with that gift card.

For this program, you can purchase the items in the specified order only. You cannot skip an item and purchase another lower in the list. If you cannot buy the current item in the list with the current card balance, then you cannot purchase any more with that card. Also, you can buy a given item only once.

Print out the number of items purchased and the remaining gift card balance.

Input: The first line (N) consists of the number of independent data sets. Each data set starts with a line containing B and C, the gift card starting balance and the number of items in the list. The next C lines have prices of each item in the list.

Output: Print out the number of items (D) purchased and the remaining balance (E) on one line. One line for each data set.

Constraints: (Balance and prices in input & output have 2 digits after the decimal point)
1<=N<=10
1.00<=B<=1000.00
1<=C<=10
0.00<=E<=B
0<=D<=C
0.01<=item prices<=10000.00

Example Input File

```
3
100.00 5
50.00
40.00
30.00
20.00
10.00
100.00 5
10.00
20.00
30.00
40.00
50.00
100.00 3
101.00
1.00
1.00
```

Output to Screen:

```
2 10.00
4 0.00
0 100.00
```

4. Gift2

Input File: gift2.dat

In the Gift1 program earlier, you tried to find how many items you can pay for with a gift card in which a list is given in order. In this program, try to find the most you can buy with the gift card. The primary goal is to buy as many items as possible. If there are two ways to buy the maximum number of items, then, among those, choose the solution, which would cost the least in total.

Print out the number of items purchased and the remaining gift card balance.

Input: The first line (N) consists of the number of independent data sets. Each data set starts with a line containing B and C, the gift card starting balance and the number of items in the list. The next C lines have prices of each item in the list.

Output: Print out the number of items (D) purchased and the remaining balance (E) on one line. One line for each data set.

Constraints:
1<=N<=10
1.00<=B<=1000.00
1<=C<=10
0.00<=E<=B
0<=D<=C
0.01<=item prices<=10000.00

Example Input File

```
4
100.00 8
50.00
40.00
30.00
20.00
10.00
5.00
4.00
2.00
10.00 2
1.00
1.00
100.00 3
101.00
1.00
1.00
100 3
30
42
41
```

Output to screen:

```
6 29.00
2 8.00
2 98.00
2 29.00
```

5. Read1

Input File: read1.dat

You want to design a program that tests how advanced your writing is. One measure of this is the size of the words in your text. In this program, you will look at word length to find how advanced your vocabulary is. Find the average word length X of a selection. Use this scale to determine if a selection is high, medium, or low: high is at least 8 letters long (X >= 8), medium is 8 > X >= 4, and low is X < 4.

In this program all text will be lowercase and all punctuation has been removed.

Input: The first line (N) consists of the number of data sets. Each data set is 3 lines long with an unknown number of words on each line.

Output: Print out "high" or "medium" or "low" for each data set.

Constraints:
1<=N<=10

Example Input File (Example has 3 sets of 3 lines each)

```
3
this line has very short words in it
it can not be very high
an ape can see high in a tree
four score and seven years ago our fathers brought forth on this
continent a new nation conceived in liberty and dedicated to the
proposition that all men are created equal
the abovementioned astrophysicist committeewoman has through
reconnaissance discovered incontrovertible evidence of
discriminatory practices that are characteristically irreproachable
```

Example Output to Screen

```
low
medium
high
```

Explanation of test cases:
In the 1st data set, there are 22 words with 69 letters for an average of 3.1 letters/word.
In the 2nd data set, there are 30 words with 143 letters, average = 4.8.
In the 3rd data set, there are 17 words with 167, average = 9.8.

6. Read2

Input File: read2.dat

The previous program uses a very basic technique to determine how hard text is to read. A more advanced algorithm is called the Flesch-Kincaid Grade Level which takes into account the total words, total syllables, and total number of sentences.

$$Level = 0.39\left(\frac{A}{B}\right) + 11.8\left(\frac{C}{A}\right) - 15.59$$

Where A = total words, B = total sentences, C = total syllables.
For example, take the sentence "The quick brown fox jumps over the lazy dog." There are 9 words, 1 sentence, and 11 total syllables (over and lazy have 2 syllables). So the reading level would be 0.39(9/1) + 11.8(11/9) – 15.59 = 2.3422… or about a 2.34 grade level (elementary school). Given a text passage, find the grade level.

The data in this program will have uppercase letters and periods ending a sentence only (no mid-sentence commas, semicolons or colons -- only periods). The number of syllables in every word will be given in the data file.

Input: The first line (N) consists of the number of unique words used in this program. Each N following lines contain the word followed by number of syllables in that word. Syllables data is not case sensitive. Hint question: Should the string comparisons be case sensitive or insensitive? The next line contains X, the number of data sets. Each data set will begin with Y, the number of lines of text, followed by Y lines of text containing an unknown number of sentences.

Constraints:
1<N<1000 (the number words in the judge data file)
1<=X<=10
1<=Y<=10

Output: Print out the reading level (to the nearest hundredth) for each data set. Round result to that precision, not truncate!

(Continued on next page…)

(Problem 6 continued)

Example Input File

```
117
a 1
abovementioned 4
against 2
ago 2
/* 113 more lines -- see data file*/
5
1
The quick brown fox jumps over the lazy dog.
3
Four score and seven years ago our fathers brought forth on this
continent a new nation conceived in liberty and dedicated to the
proposition that all men are created equal.
2
The Australian platypus is seemingly a hybrid of a mammal
and reptilian creature.
6
The abovementioned astrophysicist committeewoman has through
reconnaissance discovered incontrovertible evidence of discriminatory
practices that are characteristically irreproachable.
Indecipherable superciliousness is an incomprehensible
counterexample demonstrating indiscriminate multiplicative
presupposition.
2
This line has very short words in it.  It can not be very high.
An ape can see high in a tree.
```

Example Output to Screen

```
2.34
14.99
12.17
33.82
0.14
```

7. DandD1

Input File: dandd1.dat

I had some friends in high school who played Dungeons and Dragons®, a role-playing game (RPG) where you had characters with skills who went on adventures. A DM (dungeon master) would give a narrative of the adventure and the players would make decisions. The players would roll a dice to attack and defend. If you rolled a high number, you hit. Then you would assign damage with "hit points." The rules for combat were very complex in the old game.

For this program we will simplify it. A player has a strength value (STR) and an armor class (AC). Each weapon has an attack bonus (AB) and damage value (DAM). The attacking player rolls a 20-sided die. There are 4 possibilities:

1. A critical hit if a 19 or 20 is rolled (instant death)
2. An automatic miss if a 1 is rolled (epic fail)
3. If (roll + STR + AB) > (target's AC), then damage = (STR + DAM).
4. Otherwise, it is a miss also.

Given an attacker's stats, the defender's stats, and a 20-sided die roll, determine whether the attack is successful and assign damage if it is.

Input: The first line (N) will contain the number of data sets. Each data set consists of 3 lines. The first line is the attacker stats, 3 integers representing strength, attack bonus, and damage (STR, AB, DAM). The second is the defender stat, 1 integer representing armor class (AC). The third line consists of 3 numbers from 1-20, representing three attack rolls.

Output: For each data set, print out three lines with one of these three choices:

- `critical hit`
- `miss`
- `hit Z point(s) damage`

Separate each data set with a blank line.

Constraints:
1<=N<=10
1<=STR <= 100
1<=AB <= 100
1<=DAM <= 100

(Continued on next page…)

(Problem 7 continued)

Example Input file

```
3
5 5 5
15
1 15 19
20 20 20
20
18 2 1
1 1 1
20
1 2 20
```

Example Output to Screen

```
miss
hit 10 point(s) damage
critical hit

hit 40 point(s) damage
hit 40 point(s) damage
miss

miss
miss
critical hit
```

8. Rain

Input File: rain.dat

In this part of Washington, we are getting way more rain than average. We love it (except when it floods)! There are really cool digital weather stations that read the rainfall amount daily, transmit that data to a receiver inside your house, and then dump the water for the next days' reading. The device measures the mass of a cylinder of water, and then calculates the height of water to get the rainfall amount. The density of water is 1.0 g/cm^3 and the volume of a cylinder is $V = \pi r^2 h$. This device has a cylinder with radius 1.00 cm. 1 inch = 2.54 cm . Best to use standard Pi constants if available and needed. If not available, π is approximately 3.14159265

For example, given a mass of 3.00 g, the volume will be 3.00 cm^3. Given V = 3.00 cm^3 and r=1.00 cm, the height h = 0.9549 cm, which is 0.38 inches to the nearest hundredth.

You will write the code for the weather station to convert a mass of water (grams) into a rainfall amount (inches).

Input: The first line (N) will contain the number of data sets. Each data set consists of one number, X, the mass of the water sample in grams, rounded to the hundredth.

Output: For each data set, print out the height of water in inches, rounded to the nearest hundredth. Not truncated!

Constraints:
1<=N<=10

Example Input file
```
3
3.00
6.00
12.88
```

Example Output to Screen
```
0.38
0.75
1.61
```

9. Add

Input File: add.dat

In this problem, let's look a simple matrices, which have numbers in rows and columns. Matrices of similar dimensions can be added together. If an MxN matrix is added to another, the result will also be an MxN matrix.

For example, consider addition of these two 2 X 2 matrices.

1	2	+	1	1	=	2	3
3	4		1	1		4	5

Let's write a program to add multiple matrices with similar dimensions.

Input: The first line will contain the dimensions of the matrices (R and C) which follow in this input file. The second line (N) will contain the number of matrices. N matrices follow, each matrix is represented by R lines, representing R rows, each line/row made up of integers between -99 and 999 separated by spaces.

Output: Print out the resulting matrix for the sum of all data sets. Different column elements in a row should be separated by a space.

Constraints:
1<=N<=10
1<=R<=10
1<=C<=10

Example Input file

```
2 3
3
1 1 1
2 1 1
2 1 1
1 1 1
3 0 0
0 2 0
```

Example Output to Screen

```
6 2 2
3 4 2
```

10. Fives

Input File: fives.dat

There are many variations of the game of dominoes. When I played dominoes with my granddad, we kept score this way:
1. Add the dominoes on the end of the layout
2. You only get points (sum / 5 points), if the sum is a multiple of 5.

So if your domino total is 15, you get 3 marks.

Given a list of numbers, keep score for the domino game. You get one point for every multiple of five. No points are given if the total is not a multiple of 5.

For example, if the ends of the domino layout were 1, 4, 6, 3, 2, and 4, the total is 20. The score would be 4, since 20 is a multiple of 5 and 20 / 5 = 4.

Input: There will be an unknown number of data sets, but less than 20. Each line will contain a list of positive integers separated by a space (more than 1 number).

Output: Show the domino score for each data set on one line.

Example Input file

```
1 4 6 3 2 4
1 4 6 3 2
6 6 6 5 2
12 8 3 2 10
```

Example Output to Screen

```
4
0
5
7
```

11. Pay in Cash

Input File: cash.dat

After a tough job, Joe asks his customer for payment. Joe doesn't like credit, debit, or check. Joe likes cold, hard, cash. The problem is, whenever Joe requests his payment, customers will spend a long time sorting through their money attempting to pay in exact change. Help Joe by writing a program to see if it is possible for his customers to pay him the exact change.

Input
The first line will contain a single integer `n` that indicates the number of data sets that follow. Each data set will start with two space separated integers, x denoting the number of coins in the customer's hand, and y denoting the amount of change the customer is still trying to make. The next line will consist of x integers, representing each coin in the customers hand.

Clarification:

Coins could be of any positive integer value <= 100

Output
Depending on whether it is possible for the customer to make change of Y cents, output either `"Y is possible"` or `"Y is not possible"`

Constraints

1<=X<=30

Example Input File

```
2
10 99
25 25 25 10 10 10 5 5 1 1
10 99
25 25 10 25 10 10 1 1 1 1
```

Example Output to Screen

```
99 is not possible
99 is possible
```

12. Flipping Lights

Input File: lights.dat

Joe is working a job where he has to test many light switches. Given an initial position of a row of light bulbs and instructions on how to flip the switches, help show Joe how the row should look after he is done.

Input
The first line will contain a single integer `n` that indicates the number of data sets that follow. Each data set will start with a string representing the row of light bulbs, 1 being on and 0 off, and a single integer m representing the number of actions to be performed on the row of lights. Subsequent lines, each specify one of these six possible actions:

- FLIP A B – flips all of the lights to their inverse starting from A to B exclusive (A included but B not included)
- FLIP ALL – flips all of the lights to their inverse
- ON A B – turns on all lights starting from A to B exclusive (A included but B not included)
- ON ALL – turns on all lights
- OFF A B – turns off all lights from A to B exclusive (A included but B not included)
- OFF ALL – turns off all lights

A, B are zero indexed. For example, "ON 1 3" means, switching on second and third lights.

Output
Output what the string of lights should look like after all of the actions have been performed.

Example Input File

```
2
1010101010 4
FLIP ALL
ON 0 2
ON 8 10
OFF 4 6
000000 2
ON 0 3
FLIP ALL
```

Example Output to Screen

```
1101000111
000111
```