UTD High School Programming Contest 3–21-15
Novice Category

- **Time Allowed: three hours.**
- **Each team must only use one of UTD's computers.**
- **Answer the questions in any order.**
- **Use only Java 1.7, minGW g++, or MS Visual Studio C/C++ version 12**
- **Your program source code must be contained in one source file.**
- **Do not use the "package" construct in your Java code.**
- **Your programs must read from a named file and output to System.out (cout for C/C++ programmers.)**
- **Do not access the web.**
- **Do not use any recording device containing code, other than UTD's computer.**
- **Your solutions must be entirely yours, typed by you during the time allowed for the contest.**
- **As soon as you have solved a problem, submit ONLY the source file via your `PC^2` client.**

Scoring

Equal points will be awarded to each question, even though they may not be equally difficult.

In order to break ties, we will use penalty points. The penalty points for a question will be zero if the question is not answered correctly. If a correct submission occurs for a question at time T minutes, then T penalty points will be added, plus 20 penalty points for each incorrect submission for that question.

H. **Building a Tree**

Input File: H.txt
Run Time Limit: 10 sec

Given a sequence of characters, print a tree using those characters.

Input:
The first line of input contains n, $(1 \leq n \leq 20)$, the number of datasets to follow. Then n datasets follow, each comprising one line of text made up of printable characters from the set:

abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ*

Each dataset will contain no more than 30 characters.

Output:
Follow the format in the sample output below. The number of characters in a dataset gives the height of the tree and the number of printable characters in its base.

There is exactly one space character between each neighboring pair of printable characters in each line of the output. There should be no space characters following that last printable character in each line of output.

Sample Input	Expected Output

3
abc
AD
****
	`  a`
	` b b`
	`c c c`
	` A`
	`D D`
	`  *`
	` * *`
	`* * *`
	`* * * *`

I. **Heads and Tails**

Input File: I.txt
Run Time Limit: 10 sec

There is a simple game typically played by two players. One version of the game calls for each player to choose a unique three-coin sequence such as HEADS TAILS HEADS (HTH). A fair coin is tossed sequentially some number of times until one of the two sequences appears. The player who chose the first sequence to appear wins the game.

For this problem, you will write a program that implements a variation on this Game. You will read a sequence of 40 coin tosses and determine how many times each three-coin sequence appears. Obviously there are eight such three-coin sequences: TTT, TTH, THT, THH, HTT, HTH, HHT and HHH. Sequences may overlap. For example, if all 40 coin tosses are heads, then the sequence HHH appears 38 times.

Input:
The first line of input contains a single integer P, ($1 \leq P \leq 1000$), which is the number of data sets that follow. Each data set consists of 2 lines. The first line contains the data set number N. The second line contains the sequence of 40 coin tosses. Each toss is represented as an upper case H or an upper case T, for heads or tails, respectively. There will be no spaces on any input line.

Output:
For each data set there is one line of output. It contains the data set number followed by a single space, followed by the number of occurrences of each three-coin sequence, in the order shown above, with a space between each one. There should be a total of 9 space separated decimal integers on each output line.

Sample Input	Expected Output
4 1 HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH 2 TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT 3 HHTTTHHTTTHTHHTHHTTHTTTHHHTHTTHTTHTTTHTH 4 HTHTHHHTHHHTHTHHHHTTTHTTTTTHHTTTTHTHHHHT	1 0 0 0 0 0 0 0 38 2 38 0 0 0 0 0 0 0 3 4 7 6 4 7 4 5 1 4 6 3 4 5 3 6 5 6

J. **The Fly**

Input File: J.txt
Run Time Limit: 10 sec

Two bicycles A and B start riding toward each other at the same time from places that are 250 miles apart. A is traveling at 10 mph and B is traveling at 15 mph. At the same time, a fly leaves the front wheel of $A's$ bicycle and flies toward $B's$ bicycle at 20 mph. As soon as it reaches the front wheel of $B's$ bicycle it turns around and flies back to A. The fly continues flying back and forth, touching the front wheels of the two bicycles. It makes an infinite number of trips before being squashed between the front tires of the two bicycles, yet it travels a finite distance (the infinite series converges). How far does the fly fly?

Write a program that reads the initial distance and the three speeds and computes the distance.

input:
The first line of input contains a single integer P ($1 \leq P \leq 1000$), which gives the number of datasets to follow. Each dataset consists of a single line of input. It contains the dataset number followed by four space separated floating point values: D, the initial distance between the bicycles in miles, ($10 \leq D \leq 1000$), A, cyclist $A's$ speed in mph, ($1 \leq A \leq 30$), B, cyclist $B's$ speed in mph, ($1 \leq B \leq 30$) and F, the fly's speed in mph ($A \leq B < F$).

Output:
For each dataset output one line containing the dataset number followed by a single space, followed by the distance that the fly flies, accurate to two decimal places.

Sample Input	Expected Output
5	1 200.00
1 250 10 15 20	2 7.18
2 10.7 3.5 4.7 5.5	3 484.42
3 523.7 15.3 20.7 33.3	4 833.33
4 1000 30 30 50	5 416.67
5 500 15 15 25	

K. **Sharing the Prize**

Input File: K.txt
Run Time Limit: 10 sec

Frankie and Johnny are sweethearts. They won $100 in a programming contest and decided to go to the mall on the weekend to spend their prize money. They each started off with $50. One bought ice-creams for two, another paid for cinema tickets. One bought $20 in tokens for the video arcade, which they shared. Johnny bought sodas and Frankie bought popcorn for the movie.

At the end of the day Frankie had $23.42 left and Johnny had $12.50 left. Frankie gave Johnny $5.46 to even up their expenditures.

Your job is to write a program that will take in the prize amount to be divided between them and their expenditures during the day, and compute the amount that should change hands to even up their expenditures.

Input:
The first line of input contains a single integer P ($1 \leq P \leq 100$), which gives the number of datasets to follow. Each dataset contains several lines of data. It begins with a line containing integer $M, (1 \leq M \leq 5000)$, the amount of the prize that they must divide between them. Then comes an integer N ($1 \leq N \leq 20$) giving the number of purchases. N lines follow, each containing the name of the purchaser (either `Frankie` or `Johnny`) followed by a single space, followed by the the amount of the purchase, which will always have 2 digits after the decimal point. The expenditures of each of the sweethearts will never exceed his or her share of the prize money.

Output:
For each dataset there will be one line of output of the form `Frankie gives Johnny X` or `Johnny gives Frankie X`, where `X` is the dollar amount that must change hands to even up their expenditures. This amount must be expressed with two figures after the decimal point, even if they are both zero. If the amount needed to balance their expenditures includes a half-cent, round the value to an integer so that Frankie comes out ahead. If the expenditures of the two sweethearts match exactly print the message, "Expenditures match"

Sample Input	Expected Output
2 50 4 Frankie 5.00 Frankie 5.00 Johnny 2.00 Johnny 12.00 20 3 Frankie 5.00 Johnny 10.00 Frankie 5.00	Frankie gives Johnny 2.00 Expenditures match

L. **Fibonacci mod B**

Input File: L.txt
Run Time Limit: 10 sec

Fibonacci numbers are defined by the recurrence relation, $Fib(n) = Fib(n-1) + Fib(n-2)$, where $Fib(1) = Fib(2) = 1$.

Consider the first 10 Fibonacci numbers and those values mod 11:

n	1	2	3	4	5	6	7	8	9	10	11	12	13
$Fib(n)$	1	1	2	3	5	8	13	21	34	55	89	144	233
$Fib(n)$ *mod* 11	1	1	2	3	5	8	2	10	1	0	1	1	2

The sequence of $Fib(n)$ *mod* 11 repeats with period 10.

Write a program that calculates the period of the repeating sequence $Fib(n)$ *mod* m for different values of m.

Input:
The first line of input contains a single integer P ($1 \leq P \leq 100$), which gives the number of datasets to follow. Each dataset consists of two space-separated integers, N and M. N is the dataset number and M is the modulo value ($2 \leq M \leq 50$).

Output:
For each dataset there is one line of output. It contains the dataset number N followed by a single space, followed by the length of the repeating subsequence.

Sample Input	Expected Output
3 1 4 2 5 3 11	1 6 2 20 3 10

M. **Hailstones**

Input File: M.txt
Run Time Limit: 10 sec

The Hailstone Sequence is formed in the following way:

- if n is even, divide it by 2 to get n'
- if n is odd, multiply it by 3 and add one to get n'

It is conjectured that there is no positive starting value for n that does not reduce to 1 by repeatedly applying the above statements.

Write a program to determine the largest value in the sequence for a given n.

Input:
The first line of input contains a single integer P ($1 \leq P \leq 1000$), which is the number of datasets that follow. Each dataset consists of a single line of text containing two space-separated decimal integers. The first is the dataset number and the second is the value of n ($1 \leq n \leq 100000$),

Output:
For each dataset there is a single line of output containing the dataset number followed by a single space, followed by the largest value in the Hailstone Sequence for that starting value of n.

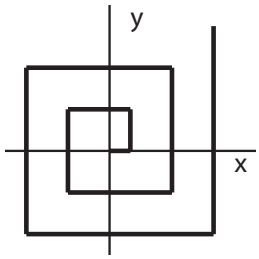
Sample Input	Expected Output
4 1 1 2 3 3 9999 4 100000	1 1 2 16 3 101248 4 100000

N. **Square Spiral**

Input File: N.txt
Run Time Limit: 10 sec

A square spiral is an infinite sequence of connected straight-line segments starting at the origin, (0,0). The first segment goes right (positive x direction). The next goes up (positive y direction) The next goes left (negative x direction), and the next goes down (negative y direction). The sequence of directions repeats.

Each segment has an integer length and each segment differs in length from the previous segment by d, a given amount. d is also the length of the first line segment. In the example below the first 10 line segments of a unit square spiral are shown. The line segments have lengths 1, 2, 3, 4, 5, 6, 7, 8, 9, and 10. A square spiral is an infinite sequence of such line segments. The spiral lines shown have endpoints in the first quadrant as follows: (1,2), (3,4), and (5,6). (We exclude endpoints on the x or y axes).



The first 10 line segments of a unit square spiral

Write a program to determine if any of the line segments in an infinite square spiral ends at a given point in the first quadrant.

Input:
The first line of input contains a single integer P ($1 \leq P \leq 1000$), which is the number of datasets that follow. Each dataset consists of a single line of text containing four space-separated decimal integers. The first integer is the dataset number. The next two integers are the x and y coordinates of the desired end point ($1 \leq x \leq 10000, 1 \leq y \leq 10000$), and the remaining integer, $d(1 \leq d \leq 50)$, gives the length of the first segment and the positive difference in lengths between neighboring segments.

Output:
For each dataset there is a single line of output. If there is no spiral solution, the line consists of the dataset number, a single space, and "NO PATH" (without the quotes). If there is a solution, the line of output consists of the dataset number, a single space, then the number of segments S in the solution ($2 \leq S \leq 1000$)

Sample Input	Expected Output
3	1 NO PATH
1 1 3 1	2 6
2 3 4 1	3 10
3 10 12 2	