UTD High School Programming Contest 3-21-2015
Advanced 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 12 C/C++**
- **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.

A. **The Fly**

Input File: A.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. How far does the fly fly before being squashed between the front tires of the two bicycles?

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	

B. **Dijkstra Maze**

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

Consider a Dijkstra Maze represented as a two dimensional array of integers, each in the range [0,9], as shown below. The maze can be traversed by taking steps between neighboring cells in any orthogonal direction (i.e., north, south, east and west). Each cell represents a cost. You have to find a minimal cost path from the top-left cell to the bottom-right cell of a given maze of size $N \times M$ where $1 \le N, M \le 30$.

0	3	1	2	9
7	3	4	9	9
1	7	5	5	3
2	**3**	**4**	**2**	**5**

Note that the solution for the given example is 24, as indicated by the bold numbers.

Input:
The input file contains several Dijkstra Mazes. The first input line contains a positive integer defining the number of Dijkstra Mazes that follow. Each Dijkstra Maze is defined by:
one line giving the number of rows, N, a single space, then the number of columns, M. N lines follow, one for each row of the maze, containing the maze numbers separated by spaces.

Output:
For each maze, output one line with the required minimum value.

Sample Input	Expected Output
2	24
4 5	15
0 3 1 2 9	
7 3 4 9 9	
1 7 5 5 3	
2 3 4 2 5	
1 6	
0 1 2 3 4 5	

C. **Family Forests**

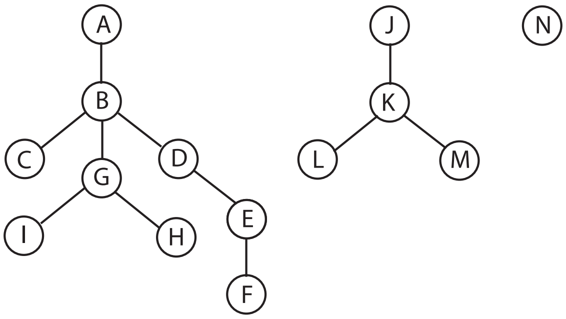
Input File: C.txt
Run Time Limit: 10 sec

Horace is trying to construct a family tree. So far he has a list names of family members going back several generations, but not many ancestry connections. He has a *forest.*

A family forest is a graph made up of vertices, one for each family member or married couple, and edges connecting pairs of vertices, one for each direct family connection. There can be no cycles in family forests and, in Horace's case, there can be many fragments that remain disconnected until he does more research. Each fragment is either a lonely vertex (an acorn) or a tree.

Write a program to read a list of the family members and the connections between them and count the number of trees and the number of acorns.

Here is an example family forest corresponding to the first sample dataset below:



Input:
The first line of the input contains a number n $(1 \leq n \leq 20)$ giving the number of datasets to follow. Each dataset consists of:

- A list of edges of the tree, (one per line, given as an unordered pair of capital letters within parentheses and terminated by a row with only a single asterisk.)
- A comma-delimited list of vertices in the tree, (these will be given on one line with a maximum of 26 vertices represented by the capital letters, A - Z).

Output:
The number of trees and the number of isolated vertices (acorns), as a sentence. For example, "`There are x tree(s) and y acorn(s).`", where x and y are the numbers of trees and acorns, respectively.

Sample Input	Expected Output
2 (A,B) (B,C) (B,D) (D,E) (E,F) (B,G) (G,H) (G,I) (J,K) (K,L) (K,M) * A,B,C,D,E,F,G,H,I,J,K,L,M,N * A,B,C	There are 2 tree(s) and 1 acorn(s). There are 0 tree(s) and 3 acorn(s).

Note: A forest may have no trees and all acorns, all trees and no acorns, or anything in between, so don't miss the forest for the trees!

D. Heads and Tails

Input File: D.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 0 0 0 0 0 0 0 38
1	2 38 0 0 0 0 0 0 0
HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH	3 4 7 6 4 7 4 5 1
2	4 6 3 4 5 3 6 5 6
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT	
3	
HHTTTHHTTTHTHHTHHTTHTTTHHHTHTTHTTHTTTHTH	
4	
HTHTHHHTHHHTHTHHHHTTTHTTTTTHHTTTTHTHHHHT	

E. **Adding Without Carries**

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

In one area of Computer Arithmetic, two numbers expressed in a certain base (or Radix) B must be added without carry propagation. For example, if $B = 2$, the carry-free sum of $X = 10111_2$ and $Y = 11011_2$ is 01100_2.

Note: subscripts on numbers denote the base used in the representations of those numbers.

Given two decimal numbers and the base, B, form their carry-free addition, base B, and convert the answer back to decimal notation.

Here is the process:

1) Write each of X and Y in base B.
2) Form each digit of the carry-free sum by adding the corresponding digits in the base B representation of X and Y.
3) Convert the result to decimal notation and print it

Base 10 modulo addition			
456_{10}	4	5	6
$+789_{10}$	7	8	9
$= 135_{10}$	1	3	5

Base 8 modulo addition				
$456_{10} = 710_8$	0	7	1	0
$+789_{10} = 1425_8$	1	4	2	5
$= 1335_8 = 733_{10}$	1	3	3	5

Write a program to compute these sums

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 is a single line that contains the data set number, followed by a space, followed by three space separated decimal integers, B, X and Y. $2 \leq B \leq 2000000$, $0 \leq X \leq 2000000$, $0 \leq Y \leq 2000000$.

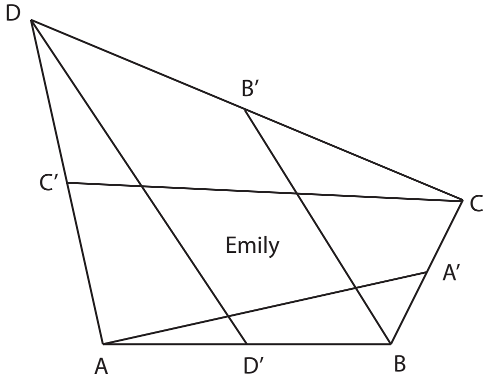
Output:
For each data set there is one line of output. It contains the data set number followed by a single space, followed by N, the decimal representation of the sum.

Sample Input	Expected Output
4	1 435
1 2 123 456	2 300
2 3 123 456	3 307
3 4 123 456	4 429
4 5 123 456	

F. **Dividing a Ranch**

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

Rancher James has a tract of land in the shape of a convex quadrilateral that he wants to divide amongst his children. Emily is to get the center part (see the diagram below)



Taking a map of the land, he marks the corners with the symbols A, B, C, D, while traversing the edges in a counter-clockwise direction. Then he connects vertex A to the midpoint of the edge BC, vertex B to the midpoint of the edge CD, vertex C to the midpoint of edge AD, and vertex D to the midpoint of edge AB. His daughter Emily will be given the center region. Calculate the area and the perimeter of Emily's tract.

Input:
The first line of input is a single integer P ($1 \leq P \leq 1000$), which gives the number of datasets to follow. Each dataset is a single line of text containing a decimal integer N giving the problem number followed by five space-separated floating point values, Bx, Cx, Cy, Dx, and Dy, in that order (where Vx and Vy are the x and y coordinates of point V given in feet). To simplify matters, A is located at $(0, 0)$ and B is located at $(x, 0)$. All coordinates will be in the range [-20000,20000].

The coordinates given will always specify a valid convex quadrilateral.

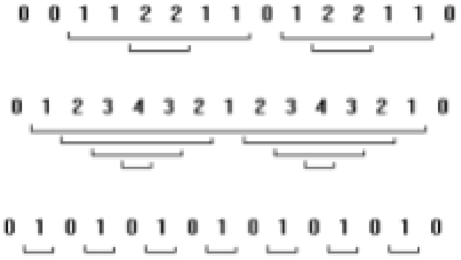
Output:
For each dataset there is a single line of output. It begins with the problem number followed by a single space, followed by the floating point value giving the area in acres of Emily's tract, accurate to three decimal places, then a single space, followed by the perimeter of Emily's tract, rounded up to the next foot. (An acre is 43,560 square feet.)

Sample Input	Expected Output
3	1 54.666 6382
1 200 250 150 -50 200	2 25.000 4589
2 200 200 100 0 100	3 29.144 4937
3 201.5 157.3 115.71 -44.2 115.71	

G. **Islands**

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

Given a sequence of integers, $a_1, a_2, a_3, \cdots a_n$ an *island* is a contiguous subsequence for which each element is greater than the elements immediately before and after the subsequence. In the examples below brackets signify the islands:



Write a program that inputs a sequence of 15 non-negative integers, in which each integer differs from the previous integer by at most one, and outputs the number of islands in the sequence.

Input:
The first line of input contains a single integer P ($1 \le P \le 1000$), which gives the number of datasets to follow. Each dataset consists of a single line of input. It contains the dataset number followed by 15 non-negative integers separated by single spaces. The first and last integers will be 0. Each integer will differ from the previous integer by at most 1.

Output:
For each data set there is one line of output containing the dataset number followed by a single space, followed by the number of islands in the sequence.

Sample Input	Expected Output
4	1 4
1 0 0 1 1 2 2 1 1 0 1 2 2 1 1 0	2 7
2 0 1 2 3 4 3 2 1 2 3 4 3 2 1 0	3 7
3 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0	4 7
4 0 1 2 3 4 5 6 7 6 5 4 3 2 1 0	