The Eighth Hunan Collegiate Programming Contest

Online Version

(with more input data and tighter time limits)

14th October, 2012
You get 14 Pages
12 Problems
&
300 Minutes

A	Three Families **Input:** Standard Input **Output:** Standard Output	

Three families share a garden. They usually clean the garden together at the end of each week, but last week, family C was on holiday, so family A spent 5 hours, family B spent 4 hours and had everything done. After coming back, family C is willing to pay $90 to the other two families. How much should family A get? You may assume both families were cleaning at the same speed.

$90/(5+4)*5=$50? No no no. Think hard. The correct answer is $60. When you figured out why, answer the following question: If family A and B spent *x* and *y* hours respectively, and family C paid $*z*, how much should family A get? It is guaranteed that both families should get non-negative integer dollars.

WARNING: Try to avoid floating-point numbers. If you really need to, be careful!

Input

The first line contains an integer *T* (*T*<=100), the number of test cases. Each test case contains three integers *x*, *y*, *z* (1<=*x*, *y*<=10, 1<=*z*<=1000).

Output

For each test case, print an integer, representing the amount of dollars that family A should get.

Sample Input	Output for Sample Input
2 5 4 90 8 4 123	60 123

Problemsetter: Rujia Liu, Special Thanks: Feng Chen, Md. Mahbubul Hasan, Youzhi Bao

B	Robot Instructions **Input:** Standard Input **Output:** Standard Output	

You have a robot standing on the origin of x axis. The robot will be given some instructions. Your task is to predict its position after executing all the instructions.

- LEFT: move one unit left (decrease p by 1, where p is the position of the robot before moving)
- RIGHT: move one unit right (increase p by 1)
- SAME AS i: perform the same action as in the i-th instruction. It is guaranteed that i is a positive integer not greater than the number of instructions before this.

Input

The first line contains the number of test cases T (T<=100). Each test case begins with an integer n (1<=n<=100), the number of instructions. Each of the following n lines contains an instruction.

Output

For each test case, print the final position of the robot. Note that after processing each test case, the robot should be reset to the origin.

Sample Input	Output for Sample Input
2 3 LEFT RIGHT SAME AS 2 5 LEFT SAME AS 1 SAME AS 2 SAME AS 1 SAME AS 4	1 -5

Problemsetter: Rujia Liu, Special Thanks: Feng Chen, Md. Mahbubul Hasan

C	Updating a Dictionary **Input:** Standard Input **Output:** Standard Output	acm

In this problem, a dictionary is collection of key-value pairs, where keys are lower-case letters, and values are non-negative integers. Given an old dictionary and a new dictionary, find out what were changed.

Each dictionary is formatting as follows:

```
{key:value,key:value,...,key:value}
```

Each key is a string of lower-case letters, and each value is a non-negative integer without leading zeros or prefix '+'. (i.e. -4, 03 and +77 are illegal). Each key will appear at most once, but keys can appear in any order.

Input

The first line contains the number of test cases *T* (*T*<=1000). Each test case contains two lines. The first line contains the old dictionary, and the second line contains the new dictionary. Each line will contain at most 100 characters and will not contain any whitespace characters. Both dictionaries could be empty.

WARNING: there are no restrictions on the lengths of each key and value in the dictionary. That means keys could be really long and values could be really large.

Output

For each test case, print the changes, formatted as follows:

- First, if there are any new keys, print '+' and then the new keys in increasing order (lexicographically), separated by commas.
- Second, if there are any removed keys, print '-' and then the removed keys in increasing order (lexicographically), separated by commas.
- Last, if there are any keys with changed value, print '*' and then these keys in increasing order (lexicographically), separated by commas.

If the two dictionaries are identical, print '`No changes`' (without quotes) instead.

Print a blank line after each test case.

Sample Input	Output for Sample Input
3 {a:3,b:4,c:10,f:6} {a:3,c:5,d:10,ee:4} {x:1,xyz:123456789123456789123456789} {xyz:123456789123456789123456789,x:1} {first:1,second:2,third:3} {third:3,second:2}	+d,ee -b,f *c No changes -first

Problemsetter: Rujia Liu, Special Thanks: Feng Chen, Md. Mahbubul Hasan

D	Searching in sqrt(n) **Input:** Standard Input **Output:** Standard Output

In binary, the square root of 2, denoted by sqrt(2), is an infinite number 1.0110101000001001111...

Given an integer *n* and a binary string (i.e. a string consisting of 0 and 1) S, your task is to find the first occurrence of S in the fraction part (i.e. the part after the decimal point) of sqrt(*n*). In case sqrt(*n*) is an integer, the fraction part is an infinite sequence of zeros.

Input

The first line contains *T* (*T*<=100), the number of test cases. Each of the following lines contains an integer *n* (2<=*n*<=1,000,000) and a binary string S with at most 20 characters.

Output

For each case, print the position of the first character in the first occurrence of S. The first digit after the dot is at position 0. The answer is guaranteed to be no greater than 100.

Sample Input	Output for Sample Input
2 2 101 1202 110011	2 58

Problemsetter: Rujia Liu, Special Thanks: Yiming Li, Feng Chen, Jane Alam Jan

E	Shortest Names	

Shortest Names

Input: Standard Input
Output: Standard Output

In a strange village, people have very long names. For example: *aaaaa*, *bbb* and *abababab*.

You see, it's very inconvenient to call a person, so people invented a good way: just call a prefix of the names. For example, if you want to call '*aaaaa*', you can call '*aaa*', because no other names start with '*aaa*'. However, you can't call '*a*', because two people's names start with '*a*'. The people in the village are smart enough to always call the shortest possible prefix. It is guaranteed that no name is a prefix of another name (as a result, no two names can be equal).

If someone in the village wants to call every person (including himself/herself) in the village exactly once, how many characters will he/she use?

Input

The first line contains T (T<=10), the number of test cases. Each test case begins with a line of one integer n(1<=n<=1000), the number of people in the village. Each of the following n lines contains a string consisting of lowercase letters, representing the name of a person. The sum of lengths of all the names in a test case does not exceed 1,000,000.

Output

For each test case, print the total number of characters needed.

Sample Input	Output for Sample Input
1 3 aaaaa bbb abababab	5

Problemsetter: Rujia Liu, Special Thanks: Yiming Li, Feng Chen, Jane Alam Jan

F	Kingdoms **Input:** Standard Input **Output:** Standard Output	

A kingdom has n cities numbered 1 to n, and some bidirectional roads connecting cities. The capital is always city 1.

After a war, all the roads of the kingdom are destroyed. The king wants to rebuild some of the roads to connect the cities, but unfortunately, the kingdom is running out of money. The total cost of rebuilding roads should not exceed K.

Given the list of m roads that can be rebuilt (other roads are severely damaged and cannot be rebuilt), the king decided to maximize the total population in the capital and all other cities that are connected (directly or indirectly) with the capital (we call it "accessible population"), can you help him?

Input

The first line of input contains a single integer T (T<=20), the number of test cases. Each test case begins with three integers n(4<=n<=16), m(1<=m<=100) and K(1<=K<=100,000). The second line contains n positive integers p_i (1<=p_i<=10,000), the population of each city. Each of the following m lines contains three positive integers u, v, c (1<=u,v<=n, 1<=c<=1000), representing a destroyed road connecting city u and v, whose rebuilding cost is c. Note that two cities can be directly connected by more than one road, but a road cannot directly connect a city and itself.

Output

For each test case, print the maximal accessible population.

Sample Input	Output for Sample Input
2 4 6 6 500 400 300 200 1 2 4 1 3 3 1 4 2 4 3 5 2 4 6 3 2 7 4 6 5 500 400 300 200 1 2 4 1 3 3 1 4 2 4 3 5 2 4 6 3 2 7	1100 1000

Problemsetter: Rujia Liu, Special Thanks: Feng Chen, Md. Mahbubul Hasan

G	Triangles in the Grid **Input:** Standard Input **Output:** Standard Output

There is a grid of $n*m$ unit squares, which has $n+1$ horizontal lines, $m+1$ vertical lines and $(n+1)(m+1)$ intersection vertices. You can choose three distinct **non-collinear** vertices to form a triangle. For example, if $n=m=1$, there are 4 vertices, which can form 4 triangles.

How many of these triangles have area between A and B (inclusive)?

Input

The first line contains the number of test cases T ($T<=25$). Each test case contains four integer n, m, A, B ($1<=n$, $m<=200$, $0<=A<B<=nm$).

Output

For each test case, print the number of triangles whose area is between A and B, inclusive.

Sample Input	Output for Sample Input
4 1 1 0 1 1 2 1 2 10 10 20 30 12 34 56 78	4 6 27492 1737488

Problemsetter: Rujia Liu, Special Thanks: Feng Chen

H

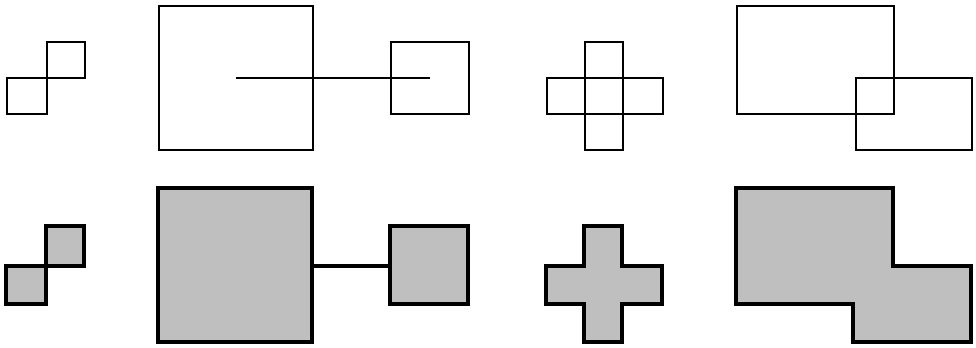
Tin Cutter II

Input: Standard Input
Output: Standard Output

In a Tin Cutting factory there is a machine for cutting parts from tin plates. It has an extraordinarily sharp knife able to make horizontal or vertical segment cuts in the tin plates. Each cutting process consists of a sequence of such cuts. Each segment cut is given by its endpoints that are always located inside the tin plate. During the cutting process some parts of tin plate can fall out and so some holes in the plate can emerge.

Factory management needs to predict the length of visible border lines at the end of the given sequence of cuts. Write a program that answers this question.

Here are four examples:



The first row in the picture contains four cuttings and the second row contains their corresponding resulting plates. Each gray area is a separate hole, and thick lines are visible border lines after cutting. There are 2, 2, 1, 1 holes respectively (from left to right), and the length of visible border lines are 8, 26, 12, 20 respectively.

Input

The first line of input contains a single integer T (T<=100), the number of test cases. The first line of each test case contains an integer n (1<=n<=100), the number of segment cuts. Each of the following n lines describe a segment cut with four integers $x1$, $y1$, $x2$, $y2$ that means a segment cut from ($x1$,$y1$) to ($x2$,$y2$) (0<=$x1$,$y1$,$x2$,$y2$<=10000). The segment is always horizontal or vertical.

Output

For each test case, print the total length of the border lines.

Sample Input	Output for Sample Input
4 6 0 0 1 0 1 0 1 2 1 2 2 2 2 2 2 1 2 1 0 1 0 1 0 0 9 0 0 4 0 4 0 4 4 4 4 0 4 0 4 0 0 6 1 8 1 8 1 8 3 8 3 6 3 6 3 6 1 2 2 7 2 8 0 1 3 1 3 1 3 2 3 2 0 2 0 2 0 1 1 0 2 0 2 0 2 3 2 3 1 3 1 3 1 0 8 0 1 4 1 4 1 4 4 4 4 0 4 0 4 0 1 3 0 6 0 6 0 6 2 6 2 3 2 3 2 3 0	8 26 12 20

Problemsetter: Rujia Liu, Special Thanks: Jane Alam Jan, Md. Mahbubul Hasan

I	# Collecting Coins **Input:** Standard Input **Output:** Standard Output	

In a maze of *r* rows and *c* columns, your task is to collect as many coins as possible.

Each square is either your start point "S"(which will become empty after you leave), an empty square ".", a coin square "C" (which will become empty after you step on this square and thus collecting the coin), a rock square "O" or an obstacle square "X".

At each step, you can move one square to the up, down, left or right. You cannot leave the maze or enter an obstacle square, but you can push each rock at most once (i.e. you can treat a rock as an obstacle square after you push it).

To push a rock, you must stand next to it. You can only push the rock along the direction you're facing, into a neighboring empty square (you can't push it outside the maze, and you can't push it to a square containing a coin). For example, if the rock is to your immediate right, you can only push it to its right neighboring square.

Find the maximal number of coins you can collect.

Input

The first line of input contains a single integer T (T<=25), the number of test cases. Each test case begins with two integers r and c (2<=r,c<=10), then followed by r lines, each with c columns. There will be at most 5 rocks and at most 10 coins in each maze.

Output

For each test case, print the maximal number of coins you can collect.

Sample Input	Output for Sample Input
3 3 4 S.OC ..O. .XCX 4 6 S.X.CC ..XOCC ...O.C ....XC 4 4 .SXC OO.C ..XX .CCC	1 6 3

Problemsetter: Rujia Liu, Special Thanks: Md. Mahbubul Hasan, Feng Chen

J Virus

Input: Standard Input
Output: Standard Output

We have a log file, which is a sequence of recorded events. Naturally, the timestamps are strictly increasing.

However, it is infected by a virus, so random records are inserted (but the order of original events is preserved). The backup log file is also infected, but since the virus is making changes randomly, the two logs are now different.

Given the two infected logs, your task is to find the longest possible original log file. Note that there might be duplicated timestamps in an infected log, but the original log file will not have duplicated timestamps.

Input

The first line contains T (T<=100), the number of test cases. Each of the following lines contains two lines, describing the two logs in the same format. Each log starts with an integer n (1<=n<=1000), the number of events in the log, which is followed by n positive integers not greater than 100,000, the timestamps of the events, in the same order as they appear in the log.

Output

For each test case, print the number of events in the longest possible original log file.

Sample Input	Output for Sample Input
1 9 1 4 2 6 3 8 5 9 1 6 2 7 6 3 5 1	3

Problemsetter: Rujia Liu, Special Thanks: Yiming Li, Jane Alam Jan

K	Cross-Shaped Tests **Input:** Standard Input **Output:** Standard Output	

There is an $n*n$ matrix. Each number can be increased or decreased. If we increase a number by x (which may be non-integer), the cost is $c*x$; If we decrease a number by x, the cost is $d*x$, where c and d are two non-negative integers.

The theoretical goal is to make every number equal to F, but in practice we only do Q tests, each test is to specify a square, then adding all the numbers in the same row or column, if the difference between the sum and $(2n-1)F$ is at most e, the test is successful.

Your task is to survive all the tests with minimal cost. It's not hard to see that the actual new matrix might be very different from the theoretical goal.

Input

The first line contains T (T<=100), the number of test cases. Each of the following lines begins with six integers n, c, d, F, e, Q(1<=n<=12, 0<=c,d<=100, -10<=F<=10, 0<=e<=5, 1<=Q<=n^2). Then an $n*n$ matrix of integers followed. Each integer will have an absolute value of no greater than 10. Then Q lines followed. Each line contains two integers x, y (1<=x,y<=n), that means we make a test on the square at row x, column y. Rows are numbered 1 to n from top to bottom, columns are numbered 1 to n from left to right. Each test case is terminated by a blank line.

Output

For each test case, print the minimal cost, to five digits after the decimal point.

Sample Input	Output for Sample Input
2 5 12 23 2 1 2 0 0 1 0 0 1 2 3 1 1 3 1 5 3 3 0 0 1 0 0 0 0 1 0 0 2 3 3 3 2 0 1 0 0 3 1 -1 -1 -2 1 1 2 1 2 2	58.00000 0.50000

Problemsetter: Rujia Liu, Special Thanks: Md. Mahbubul Hasan

L

Safe Places

Input: Standard Input
Output: Standard Output

There are *n* kinds (i.e. type-1, type-2, ..., type-*n*) of *m* satellites in the space. For each 1<=*i*<=*n*, all the type-*i* satellites are working together to protect their minimal enclosing convex polyhedron (though its volume might be zero). If a point is protected by at least *k* kinds of satellites, we say this point is safe.

Find the volume of all safe places (it might be zero).

Input

The first line contains *T* (*T*<=25), the number of test cases. Each test case begins with three integers *n*, *k* and *m*(1<=*k*<=*n*<=5, 4<=*m*<=50). Each of the following *m* lines contains an integer *t* and three **real numbers** *x*, *y*, *z*, representing a type-*t* satellite at (*x*,*y*,*z*)(1<=*t*<=*n*, 0<=*x*,*y*,*z*<=10). Each test case is terminated by a blank line

Note: The coordinates of satellites in the judge input (not sample input) are randomly generated.

Output

For each test case, print the volume rounded to 5 decimal places after the decimal point.

Sample Input

```
2
2 1 16
1 0 0 0
1 0 0 2
1 0 2 0
1 0 2 2
1 2 0 0
1 2 0 2
1 2 2 0
1 2 2 2
2 1 1 1
2 1 1 3
2 1 3 1
2 1 3 3
2 3 1 1
2 3 1 3
2 3 3 1
2 3 3 3

1 1 4
1 0 0 0
1 0 1 0
1 0 0 1
1 1 0 0
```

Output for Sample Input

```
15.00000
0.16667
```

Problemsetter: Rujia Liu, Special Thanks: Md. Mahbubul Hasan