The 41st ACM International Collegiate Programming Contest

Asia Shenyang Regional Contest

2016 年 10 月 23 日

This page intentionally left blank.

Problem A. Thickest Burger

Time limit: 1s
Color of balloons: red

ACM ICPC is launching a thick burger. The thickness (or the height) of a piece of club steak is A ($1 \le A \le 100$). The thickness (or the height) of a piece of chicken steak is B ($1 \le B \le 100$).

The chef allows to add just three pieces of meat into the burger and he does not allow to add three pieces of same type of meat. As a customer and a foodie, you want to know the maximum total thickness of a burger which you can get from the chef. Here we ignore the thickness of breads, vegetables and other seasonings.

Input

The first line is the number of test cases. For each test case, a line contains two positive integers A and B.

Output

For each test case, output a line containing the maximum total thickness of a burger.

Sample

standard input	standard output
10 68 42 1 35 25 70 59 79 65 63 46 6 28 82 92 62 43 96 37 28	178 71 165 217 193 98 192 246 235 102

Consider the first test case, since $68 + 68 + 42$ is bigger than $68 + 42 + 42$ the answer should be $68 + 68 + 42 = 178$. Similarly since $1 + 35 + 35$ is bigger than $1 + 1 + 35$, the answer of the second test case should be $1 + 35 + 35 = 71$.

This page intentionally left blank.

Problem B. Relative atomic mass

Time limit: 1s
Color of balloons: pink

Relative atomic mass is a dimensionless physical quantity, the ratio of the average mass of atoms of an element (from a single given sample or source) to $\frac{1}{12}$ of the mass of an atom of carbon-12 (known as the unified atomic mass unit).

You need to calculate the relative atomic mass of a molecule, which consists of one or several atoms. In this problem, you only need to process molecules which contain hydrogen atoms, oxygen atoms, and carbon atoms. These three types of atom are written as 'H', 'O' and 'C' repectively. For your information, the relative atomic mass of one hydrogen atom is 1, and the relative atomic mass of one oxygen atom is 16 and the relative atomic mass of one carbon atom is 12. A molecule is demonstrated as a string, of which each letter is for an atom. For example, a molecule 'HOH' contains two hydrogen atoms and one oxygen atom, therefore its relative atomic mass is $18 = 2 * 1 + 16$.

Input

The first line of input contains one integer $N (N \leq 10)$, the number of molecules.

In the next N lines, the i-th line contains a string, describing the i-th molecule. The length of each string would not exceed 10.

Output

For each molecule, output its relative atomic mass.

Sample

standard input	standard output
5 H C O HOH CHHHCHHOH	1 12 16 18 46

This page intentionally left blank.

Problem C. Recursive sequence

Time limit: 1s
Color of balloons: sky blue

Farmer John likes to play mathematics games with his N cows. Recently, they are attracted by recursive sequences. In each turn, the cows would stand in a line, while John writes two positive numbers a and b on a blackboard. And then, the cows would say their identity number one by one. The first cow says the first number a and the second says the second number b. After that, the i-th cow says the sum of twice the $(i-2)$-th number, the $(i-1)$-th number, and i^4. Now, you need to write a program to calculate the number of the N-th cow in order to check if John's cows can make it right.

Input

The first line of input contains an integer t, the number of test cases. t test cases follow.

Each case contains only one line with three numbers N, a and b where $N, a, b < 2^{31}$ as described above.

Output

For each test case, output the number of the N-th cow. This number might be very large, so you need to output it modulo 2147493647.

Sample

standard input	standard output
2 3 1 2 4 1 10	85 369

In the first case, the third number is $85 = 2*1+2+3^4$. In the second case, the third number is $93 = 2*1+1*10+3^4$ and the fourth number is $369 = 2*10+93+4^4$.

This page intentionally left blank.

Problem D. Winning an Auction

Time limit: 3s
Color of balloons: yellow

Alice and Bob play an auction game. Alice has A dollars and Bob has B dollars initially. There are N items on sale. In each round, an item will be sold by the following way. Alice writes down an integer a ($0 \leq a \leq A$) and Bob writes down an integer b ($0 \leq b \leq B$), which are the amount of dollars they want to pay for the item. If $a > b$, then Alice gets the item and pays a dollars to the seller. If $a < b$, then Bob gets the item and pays b dollars to the seller. If $a = b$, then for the 1st, 3rd, 5th, 7th ... round, Alice gets the item and pays a dollars; for the 2rd, 4th, 6th, 8th ... round, Bob gets the item and pays b dollars. Since all the items have the same value, the goal of the auction game is to get as many items as possible. Both Alice and Bob know the values of N, A and B. Your task is to calculate how many items they will get if both of them play optimally.

Input

The first line is the number of test cases. Each test case contains 3 integers N, A and B, which are no larger than 255.

Output

For each test case, output the number of items Alice and Bob will get if both of them play optimally.

Sample

standard input	standard output
3 1 1 2 2 4 2 3 3 3	Alice 0 Bob 1 Alice 1 Bob 1 Alice 2 Bob 1

This page intentionally left blank.

Problem E. Counting Cliques

Time limit:	1s
Color of balloons:	green

A clique is a complete graph, in which there is an edge between every pair of the vertices. Given a graph with N vertices and M edges, your task is to count the number of cliques with a specific size S in the graph.

Input

The first line is the number of test cases. For each test case, the first line contains 3 integers N, M and S ($N \leq 100, M \leq 1000, 2 \leq S \leq 10$), each of the following M lines contains 2 integers u and v ($1 \leq u < v \leq N$), which means there is an edge between vertices u and v. It is guaranteed that the maximum degree of the vertices is no larger than 20.

Output

For each test case, output the number of cliques with size S in the graph.

Sample

standard input	standard output
3 4 3 2 1 2 2 3 3 4 5 9 3 1 3 1 4 1 5 2 3 2 4 2 5 3 4 3 5 4 5 6 15 4 1 2 1 3 1 4 1 5 1 6 2 3 2 4 2 5 2 6 3 4 3 5 3 6 4 5 4 6 5 6	3 7 15

This page intentionally left blank.

Problem F. Similar Rotations

Time limit:	3s
Color of balloons:	orange

In mathematics especially in Euclidean geometry, we often notice rotations and corresponding rotation matrices. The dimension of the whole space of three dimensional rotation matrices is three. A natural problem is, how to measure the space. It is no doubt that the space is a metirc space if we measure the distance of two rotations as the maximum distance of two images for a point on the unit sphere.

We define the distance $dist(p, q)$ of two points p, q on the unit sphere as the length of shortest path along the surface of unit sphere. For each two rotations R_1 and R_2 one can find the point p on the unit sphere with the largest $L_{R_1,R_2} = dist(R_1(p), R_2(p))$. We define L_{R_1,R_2} as the distance of rotations R_1 and R_2.

Here we have several three dimensional rotation matrices. For each one of them, please find another one of them with the shortest distance to it in this metric space.

Input

There are no more than 100 cases. For each case, the first line consists an integer n ($1 \leq n \leq 100$), which is the number of rotation matrics. Each of the following n lines consists 9 float-point numbers $R(0, 0), R(0, 1), R(0, 2), R(1, 0), R(1, 1), R(1, 2), R(2, 0), R(2, 1), R(2, 2)$ with six decimal places corresponding to the a rotation matrix R.

Output

For each test case, output one line with n float-point numbers. The i-th one is the shortest distance to the i-th rotation from another one of them. The answer should be rounded to two decimal places.

Sample

standard input	standard output
4 1.000000 0.000000 0.000000 0.000000 1.000000 0.000000 0.000000 0.000000 1.000000 1.000000 0.000000 0.000000 0.000000 0.000000 -1.000000 0.000000 1.000000 0.000000 1.000000 0.000000 0.000000 0.000000 -1.000000 -0.000000 0.000000 0.000000 -1.000000 1.000000 0.000000 0.000000 0.000000 -0.000000 1.000000 0.000000 -1.000000 -0.000000 4 1.000000 0.000000 0.000000 0.000000 0.000000 -1.000000 0.000000 1.000000 0.000000 1.000000 0.000000 0.000000 0.000000 -0.707107 -0.707107 0.000000 0.707107 -0.707107 1.000000 0.000000 0.000000 0.000000 -1.000000 -0.000000 0.000000 0.000000 -1.000000 1.000000 0.000000 0.000000 0.000000 -0.000000 1.000000 0.000000 -1.000000 -0.000000	1.57 1.57 1.57 1.57 0.79 0.79 0.79 1.57

This page intentionally left blank.

Problem G. Do not pour out

Time limit: 1s
Color of balloons: blue

You have got a cylindrical cup. Its bottom diameter is 2 units and its height is 2 units as well.

The height of liquid level in the cup is d ($0 \leq d \leq 2$). When you incline the cup to the maximal angle such that the liquid inside has not been poured out, what is the area of the surface of the liquid?

Input

The first line is the number of test cases. For each test case, a line contains a float-point number d.

Output

For each test case, output a line containing the area of the surface rounded to 5 decimal places.

Sample

standard input	standard output
4 0 1 2 0.424413182	0.00000 4.44288 3.14159 3.51241

This page intentionally left blank.

Problem H. Guessing the Dice Roll

Time limit: 1s
Color of balloons: purple

There are N players playing a guessing game. Each player guesses a sequence consists of $\{1, 2, 3, 4, 5, 6\}$ with length L, then a dice will be rolled again and again and the roll out sequence will be recorded. The player whose guessing sequence first matches the last L rolls of the dice wins the game.

Input

The first line is the number of test cases. For each test case, the first line contains 2 integers N $(1 \leq N \leq 10)$ and L $(1 \leq L \leq 10)$. Each of the following N lines contains a guessing sequence with length L. It is guaranteed that the guessing sequences are consist of $\{1, 2, 3, 4, 5, 6\}$ and all the guessing sequences are distinct.

Output

For each test case, output a line containing the winning probability of each player with the precision of 6 digits.

Sample

standard input	standard output
3 5 1 1 2 3 4 5 6 2 1 1 2 1 3 1 4 1 5 1 6 1 4 3 1 2 3 2 3 4 3 4 5 4 5 6	0.200000 0.200000 0.200000 0.200000 0.200000 0.027778 0.194444 0.194444 0.194444 0.194444 0.194444 0.285337 0.237781 0.237781 0.239102

This page intentionally left blank.

Problem I. The Elder

Time limit:	1s
Color of balloons:	rose

Once upon a time, in the mystical continent, there is a frog kingdom, ruled by the oldest frog, the Elder. The kingdom consists of N cities, numbered from east to west. The $1-th$ city, which is located to the east of others, is the capital. Each city, except the capital, links none or several cities to the west, and exactly one city to the east.

There are some significant news happening in some cities every day. The Elder wants to know them as soon as possible. So, that is the job of journalist frogs, who run faster than any other frog. Once some tremendous news happen in a city, the journalist in that city would take the message and run to the capital. Once it reach another city, it can either continue running, or stop at that city and let another journalist to transport. The journalist frogs are too young and simple to run a long distance efficiently. As a result, it takes L^2 time for them to run through a path of length L. In addition, passing message requires P time for checking the message carefully, because any mistake in the message would make the Elder become extremely angry.

Now you are excited to receive the task to calculate the maximum time of sending a message from one of these cities to the capital.

Input

The first line of input contains an integer t, the number of test cases. t test cases follow.

For each test case, in the first line there are two integers N ($N \leq 100000$) and P ($P \leq 1000000$).

In the next $N-1$ lines, the i-th line describes the i-th road, a line with three integers u, v, w denotes an edge between the u-th city and v-th city with length $w(w \leq 100)$.

Output

For each case, output the maximum time.

Sample

standard input	standard output
3 6 10 1 2 4 2 3 5 1 4 3 4 5 3 5 6 3 6 30 1 2 4 2 3 5 1 4 3 4 5 3 5 6 3 6 50 1 2 4 2 3 5 1 4 3 4 5 3 5 6 3	51 75 81

In the second case, the best transportation time is:

- The 2-th city: $16 = 4^2$
- The 3-th city: $72 = 4^2 + 30 + 5^2$
- The 4-th city: $9 = 3^2$

- The 5-th city: $36 = (3 + 3)^2$
- The 6-th city: $75 = (3 + 3)^2 + 30 + 3^2$

Consequently, the news in the 6-th city requires most time to reach the capital.

Problem J. Query on a graph

Time limit: 2s
Color of balloons: brown

You are given a connected simple graph(in which both multiple edges and loops are disallowed) with N nodes and N edges. In this graph each node has a weight, and each edge has the same length of one unit. Define $D(u, v)$ as the distance between node u and node v. Define $S(u, k)$ as the set of nodes x which satisfy $D(u, x) \leq k$.

We will ask you to perform some instructions of the following forms.

$MODIFY\ u\ k\ d$: weight of all nodes in $S(u, k)$ increase by d or decrease by $-d$.

$QUERY\ u\ k$: ask for the sum of weight of all nodes in $S(u, k)$.

In the beginning, the weight of all nodes are 0.

Input

The first line of input contains an integer t, the number of test cases. t test cases follow. For each test case, in the first line there is an integer $N(N \leq 100000)$. The i-th line of the next N line describes the i-th edge: two integers u, v denotes an edge between u and v. In the next line, an integer $Q(Q \leq 100000)$ indicates the number of instructions. Next Q lines contain instructions $MODIFY\ u\ k\ d$ or $QUERY\ u\ k$, where $|d| \leq 100$ and $0 \leq k \leq 2$.

Output

For each MODIFY instruction, output a integer in a line.

Sample

standard input	standard output
2 6 1 2 2 3 3 4 4 1 4 5 3 6 5 MODIFY 1 1 3 MODIFY 3 1 2 MODIFY 5 2 1 QUERY 3 2 QUERY 4 1 6 1 2 2 3 3 1 1 4 2 5 3 6 5 MODIFY 3 1 5 MODIFY 2 2 2 QUERY 6 1 MODIFY 4 1 -2 QUERY 2 2	21 14 14 28

This page intentionally left blank.

Problem K. New Signal Decomposition

Time limit: 2s
Color of balloons: wine red

We consider a sequence A with p float-point numbers denoted by $a_0, a_1, ..., a_{p-1}$ where p is a prime number. To simplify our problem, we guarantee that p must be 13, 103 or 100003.

To make a decomposition for this sequence, we define the kernal functions

$$r(h,k) = 2^{sin^3\left(2\pi\frac{hk}{p}\right)}.$$

Therefore we can get a new sequence $B = \{b_0, b_1, \cdots, b_{p-1}\}$ tranformed from the original sequence A where

$$b_k = \sum_{h=0}^{p-1} a_h * r(h,k).$$

Your mission is to calculate the new sequence B.

Input

The first line is the number of test cases. Each test case contains two lines. The first line contains an integer p. The second line contains p float-point numbers corresponding to the sequence A.

Output

For each test case, output p float-point numbers rounded to three decimal places in one line corresponding to the sequence B.

Sample

standard input	standard output
13 7 0 0 0 0 0 0 0 0 0 0 0 0 13 1 2 3 4 5 6 7 8 9 10 11 12 13 13 11 7 7 7 7 7 7 7 7 7 7 7 7	7.000 7.000 7.000 7.000 7.000 7.000 7.000 7.000 7.000 7.000 7.000 7.000 7.000 91.000 85.477 92.015 93.543 91.049 99.763 98.551 98.517 97.304 106.018 103.525 105.053 111.590 95.000 102.032 102.032 102.032 102.032 102.032 102.032 102.032 102.032 102.032 102.032 102.032 102.032

Pay attention, please. You may notics that p we provided must be a prime number in $\{13, 103, 100003\}$. In order to avoid misleading, we emphasize that our standard algorithm only considers the natures of prime numbers. You may need to do some extra easy calculations offline for specified input $13, 103$ and 100003 to simplify your program.

This page intentionally left blank.

Problem L. A Random Turn Connection Game

Time limit: 1s

Color of balloons: grey

Alice and Bob play a random turn connection game in a square board with 8×8 cells. In each move, a player tosses a fair coin to decide who gets the move. That is, both Alice and Bob will get the next move with 50% chance, no matter who has moved before. Once a player gets a move, she will place a piece in an empty cell. Both Alice and Bob play randomly. That is, if there are k empty cells, each cell will be chosen with $1/k$ chance. Once Alices pieces connect the top side and the bottom side of the board, she will win the game. Similarly, once Bobs pieces connect the left side and the right side of the board, he will win the game. Pieces only connect horizontally or vertically, and cannot connect diagonally. Your task is to calculate the winning probabilities of Alice and Bob.

Input

The first line is the number of test cases. Each test case contains 8 lines and each line contains 8 characters, representing the current status of the board. The cells occupied by Alice are marked as "A", the cells occupied by Bob are marked as "B", and the empty cells are marked as ".". There is an empty line between test cases.

Output

For each test case, output the probability that Alice and Bob will win with the precision of 6 digits.

Sample

standard input	standard output
4	Alice 1.000000 Bob 0.000000
....A...	Alice 0.500000 Bob 0.500000
....A...	Alice 0.000000 Bob 0.000000
....A...	Alice 0.223093 Bob 0.198498
....A...	
....A...	
....A...	
....A...	
....A...	
....A...	
....A...	
....A...	
....A...	
BBBB.BBB	
....A...	
....A...	
....A...	
....A...	
....A...	
....A...	
BBBBA...	
...ABBBB	
...A....	
...A....	
...A....	
....	
....	
....	
....	
....	
....	
....	
.......A	

This page intentionally left blank.

Problem M. Subsequence

Time limit:	5s
Color of balloons:	white

You may notice the following program. It evaluates a kind of cost about given sequence.

$$
\begin{aligned}
&calc_{cost}(a):\\
&\quad current = 0\\
&\quad cost = 0\\
&\quad for\ (cost_0, cost_1, color) \in a\\
&\quad\quad if\ current = 0\\
&\quad\quad\quad cost = cost + cost_0\\
&\quad\quad else\\
&\quad\quad\quad cost = cost + cost_1\\
&\quad\quad current = color\\
&return\ cost
\end{aligned}
$$

For a sequence A consists of many triples $(cost_0, cost_1, color)$, its each subsequence has a cost which calculated by this program. Alice wants to know the cost of the subsequence which is the one owns the k-th largest cost.

Input

The first line of input contains an integer t, the number of test cases. t test cases follow.

For each test case, the first line consists two integers, n the length of sequence and k. The i-th line in the following n line consists three non-negative integers $cost_0, cost_1$ and $color$, where $0 \le cost_0, cost_1 \le 10000$ and $0 \le color \le 1$, corresponding to the i-th triple in the sequence.

We guarantee that the sequence has at least k non-empty subsequences.

The sum of all n is no more than 400000 and the sum of all k is no more than 400000.

Output

For each case, output the answer in one line.

Sample

standard input	standard output
1 3 4 2 1 0 1 3 1 3 1 1	3

The four sequences with the largest cost are as follow.

- $calc_{cost}(\{A_1, A_3\}) = 5$.
- $calc_{cost}(\{A_1, A_2, A_3\}) = 4$.
- $calc_{cost}(\{A_1, A_2\}) = 3$.
- $calc_{cost}(\{A_3\}) = 3$.

This page intentionally left blank.