Problem A. Bazinga!

Input file:	`lucky.in`
Output file:	`standard output`
Balloon Color:	`Blue`

Badry loves programming very much, but he also loves physics. One day, he decided to apply to work as a research assistant for Dr. Sheldon Cooper, one of the greatest minds in physics who works at the Big Bang institution.

During the interview, Dr. Cooper told Badry that he received many applications, and conducted many interviews and in this moment he is unable to decide whether to hire him or some other applicant called Kammola, because both of them are good.

That is why he decided to play the following game with Badry. First, Dr. Cooper will give each CV of his N applicants a unique label between 1 and N. Then he will shuffle them very well and put them one above the other in a random way.

Dr. Cooper labeled Badry's CV with label X and Kammola's CV with label Y, then he told him that if X is above Y then he will get the job.

Badry is now wondering about his chance in getting the job, so can you help him and tell him what is the probability of getting the job?

Note that X doesn't have to be directly above Y, meaning that other CVs can be between them.

Input

The first line of the input contains a single integer T, the number of test cases. Each test case consists of three integers N, X and Y, where $2 \le N \le 10^9$ and $1 \le X < Y \le N$.

Output

For each test case output a single number with exactly one digit to the right of the decimal point.

Example

lucky.in	standard output
1 3 1 2	0.5

Note

- Here are all the possible scenarios of three CVs where the direction left to right corresponds to up to down: [1 2 3], [1 3 2], [2 1 3], [2 3 1], [3 1 2], [3 2 1]. In three of them out of six, 1 comes above 2. That's why the answer is $3/6 = 0.5$.

Problem B. Find the path

Input file:	`path.in`
Output file:	`standard output`
Balloon Color:	`Green`

You are given a weighted connected undirected graph of N nodes and M edges (with no self-loops or multiple edges). Additionally, you have three integers u, L and K.

Your task is to find all paths of length L edges that start from node u, and for each path of them you need to sort the weights on its edges in an ascending order and report the weight in the K_{th} position, what is the maximum value among all the values you report?

Please note that an edge can appear in the same path multiple times, which means that paths don't have to be simple, read the 'Notes' section for more clarification.

Input

The first line of the input contains a single integer T the number of test cases. Each test case starts with a line containing five integers N, M, u, L and K, where $2 \leq N \leq 10^5$, $1 \leq M \leq 10^5$, $1 \leq u \leq N$, $1 \leq L \leq 10^5$ and $1 \leq K \leq L$.

The next M lines represent the edges in the graph where each line contains three integers u, v and w to indicate an edge with weight w between nodes u and v, where $1 \leq w \leq 10^9$, $1 \leq u \leq N$, $1 \leq v \leq N$ and $u \neq v$.

Output

For each test case output a single line with the maximum value.

Example

`path.in`	`standard output`
`2`	`104`
`5 5 2 7 2`	`7`
`2 1 71`	
`2 3 88`	
`2 4 50`	
`1 5 95`	
`4 3 104`	
`2 1 2 100 70`	
`2 1 7`	

Note

In the paths you can visit the same edge or node more than one time.

Problem C. Coffee

Input file:	`coffee.in`
Output file:	`standard output`
Balloon Color:	`Red`

Given the prices of different types of coffee (latte, cappuccino, $\cdots$) for each size (small, medium, and large), and a list of orders of a group of persons. Your task is to find how much will each person pay eventually. The cost that a person needs to pay is the cost of coffee he/she ordered in addition to the delivery fees. The delivery fees for each person are 100\$ divided by the number of persons (rounded down).

Eventually, the final cost should ignore 1\$ greater or less than what should be paid to the nearest multiple of 5 (i.e 44\$ and 46\$ will be rounded to 45\$. However, 47\$ and 48\$ will not be changed).

Input

The first line of the input contains a single integer T specifying the number of test cases.

Each test case begins with a line containing two integers C and P ($1 \leq C, P \leq 100$), in which C is the number of different types of coffees, and P is the number of unique persons.

Then C lines follow, giving the coffee types. Each line contains a string N and three integers S, M, and L ($1 \leq S, M, L \leq 100$), in which N is a coffee name, S, M, and L are the prices for small, medium, and large sizes, respectively. Each coffee type will appear exactly once per test case.

Then P lines follow, giving the list of orders. Each line contains three strings X, Y, and Z, in which X is a person name, Y is the coffee size ($Y \in \{small, medium, large\}$), and Z is a coffee name. It is guaranteed that all the given P names are distinct, and each person will order coffee type that exists.

Both the coffee and person names are non-empty strings consisting of lowercase and uppercase English letters with a length of no more than 15 letter.

Output

For each test case, print P lines in which each line contains two space-separated values, the name of the person and the total cost he/she will pay for his/her order. Print the persons in the same order as given in the input.

Example

coffee.in	standard output
1 3 3 cappuccino 28 34 41 latte 25 31 38 flatwhite 26 33 47 Mohammed medium cappuccino Mostafa large latte Ahmad small flatwhite	Mohammed 67 Mostafa 70 Ahmad 60

Note

The delivery fees is 100\$ and divided by 3 persons, so each of them will pay $\lfloor\frac{100}{3}\rfloor = 33\$$.

- The cost for "`Mohammed`" is 34\$ for drinks, 33\$ for delivery fees, this will be 67\$, which is the final cost.
- The cost for "`Mostafa`" is 38\$ for drinks, 33\$ for delivery fees, which will be 71\$, then 1\$ is ignored to the nearest multiple of 5, so the final cost is 70\$.

- The cost for “Ahmad” is 26$ for drinks, 33$ for delivery fees, which will be 59$, then 1$ is ignored to the nearest multiple of 5, so the final cost is 60$.