

A	Lobbying	lobbying.in	lobbying.out	2 seconds	64 MiB
B	Emergency Room	room.in	room.out	2 seconds	64 MiB
C	Social Network Vaccinations	network.in	network.out	5 seconds	64 MiB
D	Cherry Picking	cherry.in	cherry.out	2 seconds	64 MiB
E	Gadget Purchases	gadgets.in	gadgets.out	2 seconds	64 MiB
F	Health Plan Comparision	plans.in	plans.out	2 seconds	64 MiB

Problem A: Lobbying

File Name: lobbying.cpp|lobbying.java

Input File: lobbying.in

Description

As you are probably aware, any major legislation — and certainly anything on the scale of a complete reform of the US health care system — is accompanied by massive lobbying from a number of constituencies, in particular the companies that stand to profit or lose heavily depending on the outcome. Lobbying in itself need not be a bad thing: for instance, if a law about universities is debated, you would want universities, faculty, and students to get their voices heard, since many lawmakers might be unfamiliar with the particular issues at hand. It also feels at least somewhat legitimate that companies or institutions might support candidates who they think will represent them better. It gets a little more dicey when there is an element of *quid pro quo*: that concrete concessions are expected in return for financial or other support.

Perhaps one way to deal with this issue would be to allow lawmakers to accept arbitrary donations from arbitrary sources, but use knowledge of these donations to adjust their votes. If a lawmaker has heavy support from a particular constituency, then a vote in favor of the constituency is downweighted accordingly. Here, we will explore the outcome of such weighted votes.

For simplicity, we will assume that there will be a vote with two options: the status quo (which the health industry prefers), or a new health care system. You will be given, for each lawmaker, a list of all financial contributions from the health industry, with the times at which they were made. The relevant ones are the ones within the 1000 days preceding the vote. If D is the total amount of donations (in \$) the lawmaker received in that time, and he/she votes against the new system, then the vote counts as $\frac{1}{1+(D/10000)}$ votes against the new system. Any vote in favor of the new system counts fully. You are to calculate the total “amount” of vote for and against the new system.

Input

The first line contains the number K of data sets. This is followed by K data sets, each of the following form:

The first line contains three integers n, m, T . n is the number of lawmakers ($1 \leq n \leq 1000$), m the number of donations ($0 \leq m \leq 100000$), and T the time of the vote (in days). This is followed by m lines, each describing one donation. Each donation is described by two integers i, t and a floating point number d . $1 \leq i \leq n$ is the number of the lawmaker who got the donation, t the day on which it happened, and d the amount of money of the donation. A donation is relevant if $0 \leq T - t < 1000$.

Next come n lines, each describing the vote of one lawmaker. Each line contains a single character ‘Y’ or ‘N’. The character ‘Y’ means that the lawmaker voted for the reform, and the character ‘N’ that he or she voted against the reform.

Output

For each data set, output “Data Set x :” on a line by itself, where x is its number. Then, on a separate line, output the total number of fractional votes for and against the new health care system, rounded to two decimals. Each data set should be followed by an empty line.

Sample input lobbying.in

```
2
3 5 2000
1 1500 500.3
3 1999 999.2
2 2002 6822
2 1000 1000
1 1132 723.1
N
N
Y
2 0 2000
Y
N
```

Corresponding output

```
Data Set 1:
1.00 1.89

Data Set 2:
1.00 1.00
```

Problem B: Emergency Room

File Name: room.cpp|room.java

Input File: room.in

Description

The fact that many people in the US do not have health insurance causes many types of problems, including the following: Most doctors do not treat patients without insurance. Thus, obtaining medical care is difficult for those patients. However, hospital emergency rooms are required to treat all patients, because in their originally intended function, they may be the difference between life and death. But you see where this goes: since they are the only option that *must* treat uninsured patients, that's where uninsured patients will go, even for non-life-threatening cases such as flu or measles. Now, such cases will not be treated instantly, because after all, the emergency room does have life-and-death situations to deal with. Thus, what we have is people with flu or other infectious diseases spending many hours waiting in a crowded emergency room. To put it mildly, it's not clear whether that's the best strategy to prevent the spread of such infectious diseases. In this problem, we are going to explore the way in which diseases spread among emergency room visitors.

You will be given all the arrivals and departures of patients in an emergency room over time. The emergency room contains s seats, whose locations you will be given. Each arriving patient sits in the lowest-numbered seat available when he enters, and stays in that seat until he leaves. The same patient may return later, though, and will then pick a new seat. Initially, only patient 1 has the disease. A disease is transmitted from patient A to patient B if they sit at distance 2 meters or less of each other for at least 20 consecutive minutes (for instance, from time 5 until time 25; from time 5 until time 24 is not enough). If one of them leaves, then the clock starts from 0 again (i.e., it does not count to sit next to each other twice for 10 minutes each). Once a patient has the disease, he starts transmitting it the next day, i.e., 1440 minutes later; we assume that the patient stays infectious forever after that. Your goal is to figure out how many patients have the disease by the end.

Input

The first line contains the number K of data sets. This is followed by K data sets, each of the following form:

The first line contains three integers P, S, V . $1 \leq P \leq 1000$ is the number of patients (with only patient 1 sick initially), $1 \leq S \leq 100$ the number of seats in the emergency room, and $1 \leq V \leq 100000$ the number of visits of patients.

This is followed by S lines, each containing two real numbers x_i, y_i , the coordinates of the i^{th} seat in meters. Next come V lines. Each of these lines contains three integers p_j, a_j, d_j . p_j is the number of the patient making the j^{th} visit, a_j is his arrival time (in minutes, from time 0), and $d_j > a_j$ his departure time. These will be sorted by non-decreasing arrival times. Our inputs will ensure that at no time will there be more than S patients in the emergency room simultaneously. Also, if a patient i arrives at the exact time another patient j leaves, we assume that j has already given up his seat, so patient i can sit there.

Output

For each data set, output "Data Set x :" on a line by itself, where x is its number. Then, output, on a line by itself, the total number of patients sick by the end of the process. Each data set should be followed by an empty line.

(Continued on the next page.)

Sample Input/Output

Sample input room.in

Corresponding output

```
1
6 4 10
-0.5 0
0.5 1
1.5 2
-0.3 2
1 0 60
2 20 50
3 50 120
2 90 120
5 1440 1620
2 1460 1520
6 1501 1600
3 1510 1600
1 1530 1540
4 1800 2000
```

```
Data Set 1:
3
```

Problem C: Social Network Vaccinations

File Name: network.cpp|network.java

Input File: network.in

Description

Health Care policy decisions lead to or rely on many interesting computational questions. One involves vaccinations. The current approach here is to leave the decision to individuals, or, in some cases, try to vaccinate everyone. The ideal time for that is in school, as it may provide the most comprehensive access to a population. If a vaccination of everyone is overkill, it is not clear that leaving the decision to individuals is a very good policy. This approach basically assumes that the vaccinated person is the main or only one benefiting from it. But that's far from the truth: by being vaccinated, a person also contributes significantly to protecting others, for the vaccinated person cannot catch or transmit the disease. Thus, from a societal viewpoint, it would be ideal to choose vaccinated individuals carefully to protect the whole of society well.

One simple way to model this is as follows: we are given a social network (graph) with individuals and their connections. We assume the following for simplicity: whenever someone gets sick, he will infect all his unvaccinated friends, who will then also get sick. Anyone who is vaccinated never gets sick. As the public health operator, we know the number D of doses of the vaccine we have. We can use it to vaccinate D individuals. Being pessimistic, we assume that afterwards, the disease will break out at one individual so that it maximizes the number of people who eventually get sick. Our goal is thus to choose the D individuals to vaccinate to minimize the number of people who will eventually get sick, under the pessimistic assumption.

Input

The first line is the number K of input data sets, followed by the K data sets, each of the following form:

The first line contains two integers n, D . $1 \leq n \leq 30$ is the number of people in the network, and $0 \leq D \leq 6$ is the number of doses of the vaccine we have.

This is followed by n lines, each describing the friends of one individual. The i^{th} line will contain all the friends of person i , separated by spaces. Friendships are reflexive and symmetric, i.e., everyone has at least himself as a friend, and if i is friends with j , then j is also friends with i .

Output

For each data set, output "Data Set x :" on a line by itself, where x is its number. On the next line, output the number of people who will get sick in the worst case, assuming you choose an optimal set of people to vaccinate. Each data set should be followed by a blank line.

Sample Input/Output

Sample input network.in

```
1
11 2
1 2 3 4
1 4 2 3
1 2 3 4 6 5 7
7 1 2 6 5 3 4
3 4 5 6 7 8 9
3 4 5 6 7 8 9
3 4 5 6 7 8 9
5 6 7 8 9 10 11
6 5 11 10 9 8 7
8 9 10 11
9 8 11 10
```

Corresponding output

```
Data Set 1:
7
```

Problem D: Cherry Picking

File Name: cherry.cpp|cherry.java

Input File: cherry.in

Description

There was a point in time when the idea of insurance was that everyone would pay money into a joint pool, so that there would be enough money to cover one or a few individuals in case of a calamity they could not themselves afford. Of course, as participation increased, administration became more complex and more and more rules had to be created. Eventually, it was argued that such administration would be much more efficiently performed by private companies with an interest in obtaining profits. But of course, once you have a private insurance company trying to maximize profits, you get very different goals: for instance, a company will have the least interest to include those people who will need insurance the most: elderly people or patients with pre-existing conditions. It appears that it is even legal for companies to terminate policies of individuals who have paid their premiums for a long time when those individuals develop chronic (and expensive) conditions. The phenomenon of only trying to insure the most profitable individuals is also known as “cherry picking”. Here, we’ll try to develop a program to pick cherries well.

You will be given a list of n patients belonging to C categories, such as “male, 18–30”, “female, 25–40”, etc. For each individual, you will also be told what is the maximum they can afford per year for their health insurance premium, and how much money you will likely have to spend on them per year for benefits. You will be allowed to set a premium p_c for each of the categories c . Then, among the individuals in category c , those who cannot afford the premium will simply go elsewhere. The others, who can afford p_c , will pay you p_c per year, but you have to pay their benefits in return. Your goal is now to judiciously set the values to maximize your profit. However, the government is looking over your shoulder, and requiring that you insure at least a total of m people. So your pricing must be such that you also meet that requirement.

Input

The first line contains the number K of data sets. This is followed by K data sets, each of the following form:

The first line of each data set contains three integer numbers n, C, m . $1 \leq n \leq 1000$ is the number of individuals, $1 \leq C \leq 30$ the number of categories, and $m \leq n$ the government’s requirement for minimum number of insured people.

This is followed by n lines, each describing one individual i with three integers c_i, p_i, b_i . $1 \leq c_i \leq C$ is the category that individual i falls into; $p_i \geq 0$ is the maximum that individual i can pay for his/her premiums, and $b_i \geq 0$ is the amount of benefits that he/she will be receiving.

Output

For each data set, output “Data Set x :” on a line by itself, where x is its number. On the next line, output the maximum profit you can make by setting one premium per category and insuring at least m people. Each data set should be followed by one empty line.

Sample Input/Output

Sample input cherry.in

Corresponding output

```
1
8 4 5
1 300 500
1 500 400
1 600 0
3 999 0
4 1000 1500
3 1000 99273
2 50 60
2 50 70
```

```
Data Set 1:
70
```

Problem E: Gadget Purchases

File Name: `gadgets.cpp|gadgets.java`

Input File: `gadgets.in`

Description

One of the reasons why health care in the US is expensive is that many hospitals are equipped with very advanced, and expensive, machines. The cost for these machines must be recovered somehow, including by using them on a lot of patients, and charging those patients (or their insurance) for the expensive usage. Of course, there are many legitimate reasons for purchasing expensive hospital equipment, such as helping advance technology for the future to bring down the price, or actual concern for patients, even if the machine only helps a few.¹ Here, we will consider the purchasing decision as a solely economic problem, i.e., find out which machines will bring in more money than they cost.

You will be given a list of machines, each with a purchasing cost p_i , a usage cost c_i , and a maximum number of usages u_i . In addition, for each machine, you will have a suggested amount charged to the patient or insurance, r_i . This means that you pay p_i to buy the machine, and c_i each time you use it; however, the patient pays you r_i for using it. No machine can be used more than u_i times. If the demand is higher than u_i , you can only serve the first u_i patients. You will also be given a list of patients, and for each of them, which machine he or she would need to use. From this information, you should determine which machines are profitable.

Input

The first line is the number K of input data sets, followed by the K data sets, each of the following form:

The first line of each data set contains two integers n, m . $n \leq 10000$ is the number of patient visits, and $m \leq 1000$ the number of machines you are considering.

This is followed by m lines, each describing one machine by giving you the four integer numbers p_i, c_i, u_i, r_i . Finally, there are n lines, each containing one integer $m_j, 1 \leq m_j \leq m$, the number of the machine that would be needed for the j^{th} visit.

Output

For each data set, output “Data Set x :” on a line by itself, where x is its number. Then, output all the machines whose purchase would be profitable (meaning that the total revenue is strictly more than the total expenditure). The numbers should be output one per line, in sorted order. Each data set should be followed by a blank line.

Sample Input/Output

Sample input `gadgets.in`

```
1
8 4
100 0 1 150
10000 500 1000000 5600
500 100 2 300
500 100 3 300
4
2
3
2
3
4
3
4
```

Corresponding output

```
Data Set 1:
2
4
```

¹There are also less legitimate reasons, such as kickbacks from the companies producing the machines.

Problem F: Health Plan Comparison

File Name: `plans.cpp|plans.java`

Input File: `plans.in`

Description

One of the ideas of the current health care system is that the competition between private companies (as opposed to a government-run health care system) will give patients the option to choose from among many competitors the company and plan that best suits their needs. De facto, though, the market is not nearly as competitive as one would like, and often carved up geographically, with near-monopolies in many areas. In addition, there is a second problem: it is hard to know a priori which plan will be best for a person's needs, since it is hard to predict just what benefits one will need. Statistically, over large populations, predictions tend to be pretty accurate, but for just one person (e.g., you), it may be hard to say whether you will have a terrible accident in the next 10 years. If you could predict the future, choosing the right plan would be a lot easier. Although even then, figuring out from a prospectus exactly what is or is not covered isn't trivial. So you'll write a program to help with that.

You will be given texts describing different health plans, in terms of their premiums and co-payments. You will also be given your projected visits for the next year, and are supposed to figure out which of the plans will be best for you. For the following, a *sentence* always ends with a `'.'`. You can ignore all other punctuations (spaces, line breaks, commas, semicolons). The text will mix lower-case and upper-case; your program should be case-insensitive, i.e., treat them the same. Texts will contain only letters (upper and lower case), numbers, spaces and line breaks, the punctuations marks `'.'`, `','`, `','`, and the characters `'%'` and `'$'`. All numbers will be integers. The texts will be talking about:

1. A *plan*: that's just the name of the plan described. If a sentence contains the word `plan`, then the next word in that sentence is the name of the plan described after that. Until the next sentence containing the word `plan`, all information will be about that particular plan.
2. *Premium*: that's how much you pay to be in the plan. It will always be measured in dollars. In a sentence containing the word `premium`, the integer number with either a dollar sign before or after it, or the word `dollars` following it, will be the premium of the plan. (There will be only one such number.)
3. *Copayment* or *Copay*: that's what you have to pay for your visit to the doctor yourself. In a sentence containing one of the words `copayment` or `copay`, the integer number with either a dollar sign before or after it, or the word `dollars` following it, will be your maximum copayments (i.e., you will never be asked to pay more). An integer number with the percent sign or the word `percent` following it expresses copayments as a percentage of the actual cost. If the description of a plan contains both a dollar amount and a percentage for copayment, then this means the *minimum* of the two corresponding amounts. The copay can never exceed the cost of the visit itself. That is, if the copay is \$100, and the visit costs \$50, then you only pay \$50.
4. *per visit* vs. *total*: this specifies whether co-payments described in this sentence apply *for each visit separately*, or *for all visits together*, depending on which of the words `per visit` or `total` appears in the sentence. If neither appears, then the sentence applies for total copayments. If both appear, then it applies per visit. (Obviously, for the premium, there is no "per visit" vs. "total".)
5. Each sentence will contain at most one of the words `premium` or `copayment/copay`. Also, each sentence will contain the word `plan` at most once.
6. If the premium or copayment are not specified, they are 0.
7. If different values for the premium or copayment of a plan are specified, then the last one specified should be used.

After being given the description of the plans, you will be given a list of visits to the doctor. Each has just one number, namely, the cost of that visit. Your goal is then to list the total expense for each plan.

(Continued on the next page.)

Input

The first line contains the number K of data sets. This is followed by K data sets, each of the following form:

The first line contains two integers m, n between 1 and 1000. This is followed by m lines of text describing the health plans. Each line has at most 80 characters. There will always be at least one plan. There can be multiple sentences on a line, and sentences can span multiple lines. Next come n lines, each giving one integer, the cost of that visit to the doctor's.

Output

For each data set, first output "Data Set x :" on a line by itself, where x is its number. Then, output, one per line, the total amount of money you have to spend (for premium and copay) for each of the plans, rounded to two decimals. These should be ordered in the order that the plans are *first mentioned* in the text. Each data set should be followed by a blank line.

Sample Input/Output

Sample input plans.in

Corresponding output

```
1
9 5
In this article, we will compare a number of different health
insurance options. We will look at 3 different plans.
Plan UnNeeded is intended for healthy individuals; it has a low
premium of 500 dollars.
Plan MiddleOfTheRoad offers a balanced option. Its premium is $2000.
The copayment per visit is 10% or 200$, whichever is lower.
Plan Sicko is intended for heavy usage. Its premium is $3000. The
copayment is at most 10$ PER VISIT or 3 percent.
Oh, we also forgot to mention; Plan unneeded has a total copy of 10%.
500
110
1750
2000
12000
```

```
Data Set 1:
2136.00
2636.00
3043.30
```