South German Winter Contest

January 23, 2010

The Problem Set

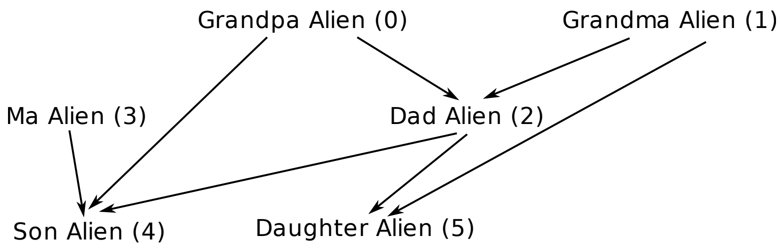
No	Title	Page
A	Alien Genealogy	3
B	Breaking the Code	5
C	Clones	7
D	Heisenberg Uncertainty Principle	9
E	Invasion Force	11
F	Monsters on Train Trip	13
G	Self Replicating Worms	15
H	Shields	17
I	Space Slugs	19
J	Susan likes Skating	21
K	Team Triathlon: Monsters vs. Aliens	23

Good luck and have fun!

This page is intentionally left (almost) blank.

Problem A
Alien Genealogy

Aliens are strange people. In contrast to us humans they may have more than two parents and in fact it is also quite common that their grand parents, grand-grand parents, ... are involved in their procreation (did I mention that aliens are strange?). Anyway, this tradition is reflected in the families' genealogies which look quite strange to us (the numbers next to the names correspond to the numbers in the sample in-/output below):



Since this tends to irritate quite a lot of humans, the intergalactical alien council decided to revise all genealogies so that they look more familiar to what humans are used to. For example, in the small cutout from an alien genealogy depicted above the connection between Grandpa Alien and Son Alien should be omitted in the revised version because Grandpa's genes are already present via the connection from Dad Alien. Also Daughter Alien already inherited some of Grandma's genes from Dad Alien so the direct link from Grandma can be left out.

Your task is two write a program for the intergalactical alien council that takes all genealogies and simplifies them as far as possible so that the inheritance of genes stays correct. Fortunately, aliens are still somewhat natural beings and cannot betray the laws of physics so there are no "loops" in their genealogies.

Input

The input starts with the number of genealogies $1 \leq n \leq 100$ in the first line. Each of the following n genealogies starts with the number of links $1 \leq k \leq 4\,950$ and the number of involved aliens $1 \leq l \leq 100$ in the first line, separated by white space. Then k lines containing the links follow. Each link consists of two names (for simplicity they are all encoded as numbers between 0 and $l - 1$).

Output

After simplifying each genealogy, output its links in the same order as in the input but leave out the superfluous ones. Separate each pair of genealogies by a blank line.

Sample Input

```
1
8 6
0 2
1 2
2 5
2 4
1 5
3 4
3 5
0 4
```

Sample Output

```
0 2
1 2
2 5
2 4
3 4
3 5
```

This page is intentionally left (almost) blank.

Problem B
Breaking the Code

The evil alien Gallaxhar manoeuvred his giant spaceship to our galaxy. There is little left that could stop him. His spaceship is protected by an energy field that withstands all human weapons and also makes it impossible for anyone to approach the ship. So the mission to rescue the world would be doomed if it wasn't for you, Dr. Cockroach, the mastermind of the monsters.

Since Gallaxhar's supercomputer is much more powerful than all human computers together, he relied on a very simple but extensive code for the energy field and didn't even mind keeping it secret. Your time is too limited to build a coequal computer yourself so you have to restrict to the simple personal computer you're sitting at in this moment. So all comes down to one question: Can you outwit the code and shortcut its computation?

The code works as follows. Starting from an arbitrary number x in the range from 0 to $n-1$, one step is to compute the code function

$$f(y) = (((y + a) \cdot b)^c) \bmod n$$

where u mod n returns the unique number v in the range from 0 to $n-1$ such that $u - v = d \cdot n$ for some integer d (a, b, c, d, n, u, v, y are all integral). Gallaxhar knows that this isn't hard enough. Thus, he iterates this process q times, i.e. the code is required to calculate $f^q(y)$ where $f^0(y) = y$, $f^1(y) = f(y)$, $f^2(y) = f(f(y))$, $f^3(y) = f(f(f(y)))$ and so on. As if this wasn't hard enough, he adds another step, the inversion of this function. So your task is to find the unique number y in the range from 0 to $n-1$ such that, for given x,

$$x = f^q(y).$$

This problem seems very hard, since Gallaxhar uses huge numbers for all parameters. So good luck!

Input

The first line holds the four parameters of the code function f: n, a, b, c with $1 \leq n \leq 10^6$, $-10^{18} \leq a, b \leq 10^{18}$ and $0 \leq c \leq 10^{18}$. The second line contains the number of function values k which you have to invert ($0 \leq k \leq 10^6$) and the number of function applications q ($0 \leq q \leq 10^{18}$). Then k lines follow, each containing one value x.

As a reminder, all numbers are integral.

Output

For each value of x of the input, output the unique value y in the range from 0 to $n-1$ such that $x = f^q(y)$ on a single line.

Sample Input

```
10 1 1 1
3 2
3
7
1
```

Sample Output

```
1
5
9
```

This page is intentionally left (almost) blank.

Problem C
Clones

The alien Gallaxhar tries to defeat the monsters by cloning himself many, many times. The monsters dressed up to look like one of these clones and try to hide among them. The clone army (and the disguised monsters) march in a rectangular grid (see figure below). If a clone spots a monster at another place in the grid, he tries to kill the monster with a laser weapon. However, it only hits the monster if there is no other person in the direct line in between.

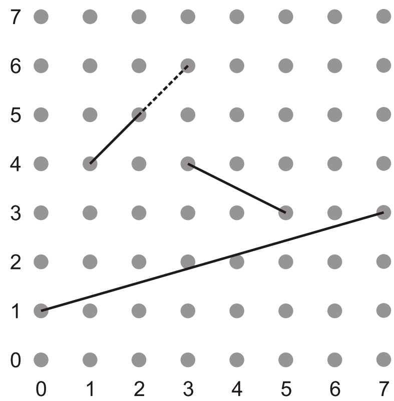


Figure 1 – Rectangular grid of 64 persons, showing the first four test cases of the sample input.

Input

The input consists of several test cases. The first line of the input contains an integer c ($1 \leq c \leq 100$), giving the number of test cases. c lines follow, each containing the coordinates of the clone and the monster. Each location is given by two integers x and y ($0 \leq x, y < 10^{50}$).

Output

Print one line for each pair of persons. Print "`Hit!`" if there is no other person in the direct line of the clone and the monster, "`Miss`" otherwise.

Sample Input

```
6
3 4 5 3
7 3 0 1
1 4 2 5
1 4 3 6
47 11 1356 1286
47 11 1356 1288
```

Sample Output

```
Hit!
Hit!
Hit!
Miss
Miss
Hit!
```

This page is intentionally left (almost) blank.

Problem D
Heisenberg Uncertainty Principle

Maybe you have heard of the *Heisenberg Uncertainty Principle* before. It states that some pairs of physical properties, like position and momentum, cannot both be known with arbitrary precision. This principle can be extended for the prediction of the appearance of aliens and monsters. You may predict either the location **or** the time of the appearance exactly. The other property is only known with a certain fuzziness.

Currently, you are searching for a habitable planet outside of your solar system. One characteristic number for habitable planets is the chance of a simultaneous appearance of aliens and monsters (less is better). We already tried to predict the locations and times of the appearances. But as you now know, we cannot get those two properties simultaneously. However, we are able to predict the exact location for every alien and the exact time for every monster. Please help us to determine the number of possible clashes between aliens and monsters.

Input

The first line of the input gives the number of planets ($0 < p < 100$) that need to be investigated. Each of the planet descriptions starts with two numbers on one line: a and m, the number of aliens and monsters, respectively ($0 < a, m < 20\,000$). Then follows a lines, three numbers $starttime_i$, $endtime_i$, and $location_i$[1] in each line representing the fuzzy timespan and the exact location for the ith alien ($starttime_i < endtime_i$). Afterwards follow m lines, three numbers $startloc_j$, $endloc_j$, and $time_j$ in each line representing the fuzzy location and the exact timespan for the jth monster ($startloc_j < endloc_j$). All times and locations are nonnegative integers less than 10 000 000.

Output

For each planet, print one line containing the number of possible conflicts between aliens and monsters. You may assume that we have already filtered out the worst planets, i.e. there is no planet with more than 55 555 conflicts.

Sample Input

```
2
3 1
2 10 5
5 15 5
11 19 5
1 10 10
2 2
1 7 12
2 23 3
5 16 6
1 42 2
```

Sample Output

```
2
3
```

[1]Recently, a clever scientist found out how to map the three dimensions of space down to exactly one dimension.

This page is intentionally left (almost) blank.

Problem E
Invasion Force

An alien force is trying to invade the earth as usual. After some failed attempts in the past by kidnapping single persons, mutilating cows or fixing soccer matches they try a new strategy. They developed this strategy by watching the humans closely. It will consist of arming their ships with soldiers, fly to earth and kill people there.
While flying and killing seem to be the easy part, the problem arises in selecting the soldiers for their mission. The aliens have chartered old space buses with a limited number of passenger seats. As each alien has a unique shape (you might have seen this in the movies), each soldier will occupy a certain number of seats. Furthermore, the combat power of the different aliens vary a lot, because some have lots of arms to carry lots of laser guns.
The alien general wants as much combat power as possible that will fit in the rented space bus. The combat power of a group of aliens is the sum of all individual combat power values. As this task is way too annoying for the general of the mission, they kidnapped you to solve the problem. They promised not to eat you, of course.

Input

The first line of the input holds the number t of test cases ($0 < t < 10$) that follow. Each test case consists of multiple lines. The first line of each test case holds two numbers, the number of soldiers n ($0 < n < 1\,000$) and the number of seats s in the space bus ($0 < s < 1\,000$). You can assume that the seats are in one long line, so it is easy to place the aliens. Then follow n lines describing each alien soldier of the group with two entries. The first is number of seats s_i the alien requires ($0 < s_i \leq s$) and the second one is the corresponding combat power c_i ($0 < c_i < 10\,000$). Higher values for c_i means stronger combat power.

Output

For each test case, print the maximum of possible combat power that fits into the given space ship onto a single line.

Sample Input

```
2
3 4
2 1
2 1
2 1
5 8
1 1
5 2
4 3
7 5
4 4
```

Sample Output

```
2
7
```

This page is intentionally left (almost) blank.

Problem F

Monsters on Train Trip

The monsters are superheros with a big organisation now. They have rescue operations all over the world. They like to travel to their site of operation by train. After some time they developed sympathy and antipathy to some other members. The sympathy/antipathy relations with certain members of the team and their presence in the same wagon affects their performance in superheroing. Every monster has some benefit for a operation. This benefit is influenced by the presence of other monsters while traveling. Some monsters are only influenced if a specific set of monsters is present in their wagon. For example small monsters get pissed, if two huge monsters talk all the time about problems being such huge.

Theoretically, one train wagon would be large enough to transport all monsters. However, they reserve up to three wagons to increase the benefit for the operation. Most important in the assignment of monsters to wagons is to increase the total benefit for the operation. The number of used wagons comes second (less is better). At least one monster must take part in every operation.

Input

The inputs starts with the number of operations (at most 10). Each operation is described as follows. It starts with the number N of monsters, which may take part in this operation, in the first line ($0 < N \leq 10$), followed by N monster descriptions. Every monster description starts with one line containing his name (at most 20 alphanumeric characters), its base benefit B ($-100 \leq B \leq 100$) for the operation, the number of antipathy/sympathy sets M ($0 \leq M \leq 10$). Each of the next M lines consist of one antipathy/sympathy set. A set consists of the benefit increase/decrease C ($-100 \leq C \leq 100$) by this set, followed by the number of involved monsters I ($0 < I < N$) and I names of monsters in this set. If every monster in this set is in the same wagon of the given monster, his benefit is increased/decreased by C. Multiple increases/decreases can be applied to one monster.

Output

For every operation, output the maximum benefit you can achieve choosing a set of monsters and the number of used wagons on one line.

Sample Input

```
1
3
Ginormica 2 0
Insectosaurus 2 0
MissingLink 2 1
-2 2 Ginormica Insectosaurus
```

Sample Output

```
6 2
```

This page is intentionally left (almost) blank.

Problem G
Self Replicating Worms

When the quantonium fell on Susan, also an innocent family of earthworms was hit by a little chip of the meteorite. The earthworms collapsed into tiny little pieces. The different pieces started replicating immediately. Depending on what kind of piece was renewing itself new pieces appeared and substituted the old piece. After a while the new pieces replicated again, disappeared and were substituted by the new parts. So the tiny little earthworm pieces grew bigger and bigger and new dangerous worms suddenly creeped on earth.
Let's take a look at the most dangerous of these new worms (see sample below). They started out from the former bowel piece of the innocent earthworms. The bowel piece either changed into an radioactive piece attached to a electric piece or it changed into a muscle piece and a spider piece.

> bowel → electric radioactive
> bowel → muscle spider

These four new pieces also replicate:

> electric → radioactive long
> radioactive → muscle electric
> muscle → spider flexible
> spider → muscle bowel

Strangely some parts (e.g. the long and flexible pieces) were not replicating. Using this replication mechanism a new worm can evolve as:

> bowel
> ⇒ electric radioactive
> ⇒ radioactive long muscle electric
> ⇒ muscle electric long spider flexible radioactive long

This worm sounds pretty scary, doesn't it? Also the other worm parts started extending themselves. They use the same mechanism. There are the following replications:

> tail → spider muscle
> head → head electric

But for some unknown reason only the new worms starting from the bowel are real monsters. All other worms are just gentle new creatures living under the surface. So General W.R. Monger is desperately trying to find the bowel worms. He managed to decode the replication mechanism but struggles to tell whether a worm started from the bowel. Can you help him?

Input

The input consists of two parts. First the replication rules are given, and second, the worms are given based on the sequence of their pieces. The replication part starts with an integer in the first line stating how many replication rules r follow ($1 \leq r \leq 20$). Each replication is written in a separate line. A replication starts with one piece, which has to be replaced, followed by the names of the two pieces it can be replaced with. One piece can only be replaced by two other pieces. In the next line an integer is given with the number of worms w ($1 \leq w \leq 30$) following. Each worm (at most 128 pieces) is written in one line. Remember that the start piece we are interested in is always `bowel`. The worm pieces consist of a maximum of 100 alphanumeric letters.

Output

For each worm given in the input output one line containing "`Yes`" if this worm replicated from a bowel with the help of the given replication mechanism, "`No`" otherwise.

(Sample Input and Output are provided on the next page.)

Sample Input

```
8
bowel electric radioactive
bowel muscle spider
electric radioactive long
radioactive muscle electric
muscle spider flexible
spider muscle bowel
tail spider muscle
head head electric
5
muscle electric long spider flexible radioactive long
electric radioactive
muscle radioactive
muscle head
head radioactive long
```

Sample Output

```
Yes
Yes
No
No
No
```

Problem H
Shields

As monsters and aliens fight each other again and again on our planet, some of us started building large shields. In case of the next fight we can protect us behind these shields. Each shield has its own properties and can protect a certain number of persons. However, some of the shields do not work reliable. Thus, we are a little bit scared and do some calculations about our safety.
We defined the *Unneeded Set of Shields* (USS) as any set of shields that are not really necessary to protect the requested number of persons. There are (hopefully) many of those USS. To characterize the reliability of our shield network, two numbers are interesting:

- *Largest Unneeded Set Count* (LUSC): the number of shields in the largest USS.
- *Guaranteed Unneeded Set Count* (GUSC): it must be guaranteed that any set with less or equal than GUSC shields is a USS; we are interested in the maximal possible GUSC number.

Input

The first line of the input gives the number of test cases ($0 < c < 100$). Each test case description starts with two numbers on line: the number of shields ($0 < s \leq 1\,000\,000$) and the number of persons to protect ($0 < p \leq 1\,000\,000$). The second line contains s numbers. The ith number on that line describes how many persons p_i are protected by the ith shield ($0 < p_i \leq 1\,000\,000$).

Output

For each test case, print LUSC and GUSC on one line.

Sample Input

```
3
4 10
3 4 5 6
3 10
5 2 3
9 1000
11 670 24 113 188 422 547 823 230
```

Sample Output

```
2 1
0 0
7 2
```

This page is intentionally left (almost) blank.

Problem I
Space Slugs

The alien race of the Xyth is expanding its population to the planet Alrini. The planet consists of rocky mountains that are perforated by caves and tunnels. As the Xyth are rather squashy they can not live on bare rock due to the danger of ripping and damaging their sensitive skin. Being a highly developed race, they of course have a solution for this problem.

To make the caves inhabitable they dump space slugs imported from planet Aicipici from their massive space ships. These stupid slugs are huge beasts being several meters in diameter and up to 50 meters long. Being a rather simple organism, they can be easily controlled by neuro-implants and steered in the desired direction. Just like their smaller earthly relatives, the space slugs are extremely slimy, which is perfect for the Xyth, as the slime covers the rock of the tunnels and caves and makes them cozy and inhabitable. Unfortunately, due to the large size and weight of the slugs, they can only go downward in the steep caves of planet Alrini. So, to cover all tunnels and caves of the cave system with slime, multiple slugs have to be dropped at its entries. As the transportation of the giant space slugs is quite expensive, the Xyth need to minimize the number of slugs used. Fortunately, the geologists of the Xyth already created maps of the caves and tunnels. Interestingly, the maps showed that for any two caves there is at most one way to travel between them without going to the planet's surface.

Input

The first line of the input consists of two positive numbers, the number of caves $n \leq 1000$ and the number of tunnels $m \leq 1000$. The caves are sorted by height and named by the numbers from 1 to n; 1 is the highest cave, n is the lowest one, and the cave i is strictly higher than cave $i + 1$. The following m lines contain two numbers each which give the indices of two caves connected by a tunnel. There may be at most one tunnel connecting any two caves. All numbers in the same line are separated by a single space.

Output

Print the minimal number of slugs needed to cover all tunnels and caves in slime followed by a single line break. Slugs may be dropped at any cave, as the caves are all connected to the surface.

Sample Input

```
7 5
1 2
2 3
2 4
4 5
5 7
```

Sample Output

```
3
```

This page is intentionally left (almost) blank.

Problem J
Susan likes Skating

Susan alias Gigantica definitely likes skating. In her gigantic form she likes to put a car under each foot and use them as skates. However, if there are other things on the road (like people or lamp posts) they might get destroyed or at least harmed. After their victory over the aliens, it is the latest trend to treat monsters like superheroes. Thus, the mayor cannot limit her freedom to move around the city. Instead he wants to suggest to her that she limits her skating-tours to a small set of roads. His only problem is now that he cannot figure out, which set he should suggest to her: she should still be able to reach every point of the city, but the amount of endangered objects (and humans!) should be minimized. Of course, one-way-roads do not matter for a monster, so she will use roads in both directions. Dr. Cockroach already tried to help, but somehow all he did was making an explosion. Now the mayor is relying on your help!

Input

Input consists of a single test case. Input starts with two numbers n, m on a single line ($2 \leq n \leq 1000$; $n-1 \leq m \leq n \cdot (n-1)/2$). n denotes the number of places/junctions, m the number of streets between these places. Places are numbered 0 to $n-1$ and not further specified. Input continues with m lines, one line per street containing three integer numbers s, e, c, the start- and end-point of the street and the number of endangered objects ($0 \leq s, e \leq n-1$; $1 \leq c \leq 200$). There is always a set of routes that enables Susan to reach every other point of the city from every point.

Output

Output a single line containing the minimized number of endangered objects and humans (accumulated).

Sample Input

```
4 6
0 1 3
0 2 3
0 3 3
1 2 1
2 3 2
3 1 1
```

Sample Output

```
5
```

This page is intentionally left (almost) blank.

Problem K

Team Triathlon: Monsters vs. Aliens

As you might have noticed throughout the remaining problems of this contest not very many Aliens and Monster believe in peaceful coexistence and harmony amongst each others. But those few who do eventually got together at the ACM (Aliens Competing Monsters) Congress and decided to strengthen the understanding between their cultures with a sports event: The Monster vs. Aliens Team Triathlon!

A team triathlon differs a bit from a regular triathlon: Instead of one creature performing all three disciplines in a row three creatures share the disciplines in form of a relay (i.e., the second waits in a transition area for the first to finish its discipline while the third waits for the second before starting to enter the competition). Exactly one team of Monsters will compete against exactly one team of Aliens in each round.

The disciplines are as follows: One crater length of quicksand swimming on the companion of Betelgeuse, a star-to-star speedrace on a muscular-powered spaceship and finally moon-running of a distance of twice the Golden Gate Bride (chosen since it is one of the most favorite objects to be destroyed by Aliens) on one of the 63 satellites of Jupiter.

There is a little problem with the peaceful and harmonic concept of this sports event: Monsters and Aliens don't trust each other (yet) in determining the winning team. Hence they ask you to be their incorruptible referee and would like you to create a reliable program returning the winner, i.e., the team that needs less time to complete the three disciplines.

Input

The input starts with an integer c ($0 < c < 100$), the number of rounds you are supposed to judge. Each round is given by one line of six whitespace separated integers $q_M, s_M, r_M, q_A, s_A, r_A$ ($0 < q_M, s_M, r_M, q_A, s_A, r_A < 10\,000$), the amount of time the Monster (or Alien) needs for quicksand swimming, spaceship racing or moon-running.

Output

For each round output one string, either "`Monsters`" (if the Monster win the triathlon), "`Aliens`" (if the Aliens win the triathlon) or "`Complimentary Slime for Everyone`" (if Monsters and Aliens are tied).

Sample Input

```
3
1 2 3 9 8 7
2 3 4 3 2 4
8 9 7 3 1 2
```

Sample Output

```
Monsters
Complimentary Slime for Everyone
Aliens
```

This page is intentionally left (almost) blank.