# 12th Iran Internet Programming Contest

Sponsored by University of Tehran - College of Engineering

*Hosted by ShareCode*

# A– **Davood Inchi Code**

## Description

Bahman loves Dan Brown's books. He has read all of them and now he thinks he can come up with a nice story himself. This is part of his drafts for the book:

> "... Rambod Langaan found the dead body on the floor. He couldn't see clearly in that dark, but he could definitely recognize the smell. The body should have been there for a few days. Holding his breath, he got closer and reached the bag. It was wet.
> As soon as he got out of the tunnel, he filled his lungs with fresh air. Now he could think more clearly. The bag was wet; not from the water in the tunnel, but from the blood. His body shivered. He had no choice. Without any further thoughts, he opened the bag and immediately found what he was looking for: an old piece of parchment. He couldn't believe what he was seeing. Those were the exact same notes that Davood Inchi wrote more than a thousand years ago. They were the rules of their underground society, called Folan, but they were what all people thought. Rambod knew more than those. This was actually an encrypted message for those who were really a part of Folan, the location of their secret gatherings. Taking out the characters at Fibonacci indices and putting them together would reveal the message. That was easy for Rambod. He wrote the first, second, third, fifth and eighth characters out and wrote them on his hand; "Thefsl" was what he had on his hand! What could that possibly mean? The sentence on the parchment started with "The first rule". "This cannot be right" he thought. He tried more letters but suddenly he reached the end of the parchment. But there were not enough letters. So he continued backwards through the text. He reached the beginning of the text, reversed the direction and continued picking toward the end of the text. He couldn't do this forever. He had no time. There was someone out there killing people... "

Forget about what Bahman was doing! Just help Rambod Langaan to find the hidden message.
And don't try to find the secret place! You can't!
What you should do is to pick characters with Fibonacci steps which are 1, 1, 2, 3, ... (so pick the first, second, third, fifth, eighth and ... characters). If you reach one of the end-points in the string, then reverse your direction and continue. Moreover, if you where to pick a '-' character, reset your Fibonacci sequence to 1, 1, 2, 3, ... . Continue until you have enough characters.

## Input

The input contains several test cases.
In the first line of input comes $T$ $(0 < T < 64)$, the number of test cases.
Each test case consists of two lines, a line with a number $K$ $(0 < K < 32)$, denoting the number of characters Rambod needs to find, and a line with a string with length $l$ $(1 \leq l \leq 1024)$, the text on the parchment. The string includes at least one character other than '-'.

## Output

For each test case write the hidden message on a separate line.

## Sample Input

```
2
6
Hello World
15
12th-Iran-Internet-ACM-ICPC-2014-Final
```

## Sample Output

```
Heloor
12tIraIntrtICP2
```

# B– **Persian Map**

## Description

Recently, archaeologists found some maps of Iran (formerly known as Persia). These maps belong to different dynasties and contain the name of the king of that time on top, like Keyumars, Bahram Chobin, Hushang, Jamshid, Kai Khosrow and many others. On each map there are some roads, each connecting exactly two cities. Some roads didn't exist at a certain period of time (when they were not built yet or the road was covered in sand, blocked by a pile of dead soldiers who died in a great battle or washed away by a flood), so they might not exist in some maps. The good news is that today with the help of technology, archaeologists can dig the ground to find those roads. We know for sure that if at some point of time there existed some road between two cities, any road mentioned in earlier or later maps between the same cities is the same road. We now want to help the archaeologists find out how many roads can possibly be found. You should merge the maps (add all the roads that existed on at least one map without adding the redundant roads) then count the number of roads.

## Input

The input contains several test cases.
In the first line of input comes $T$ $(0 < T < 100)$, the number of test cases.
Each test case starts with $n$ $(0 < n < 10)$ and $m$ $(0 < m < 1000)$, the number of maps that the archaeologists found and the number of cities in Iran, respectively. Then comes the data for each map.
The map starts with the name of the king (without any spaces, at most 64 characters in length) and $R$ $(0 \leq R < 1000)$, the number of roads in that map.
Then follows $R$ lines. Line $i$ contains two distinct integers $x$ and $y$ $(0 \leq x, y < m)$, the cities road $i$ connects.

## Output

For each test case, print a separate line containing the total number of roads that the archaeologists can find.

## Sample Input

```
2
3 5
Hushang 3
0 1
1 2
2 3
Keyumars 4
0 1
1 3
2 1
4 3
KaiKhosrow 3
1 0
2 1
3 2
1 10
Jamshid 9
0 1
1 2
2 3
3 4
4 5
5 6
6 7
7 8
8 9
```

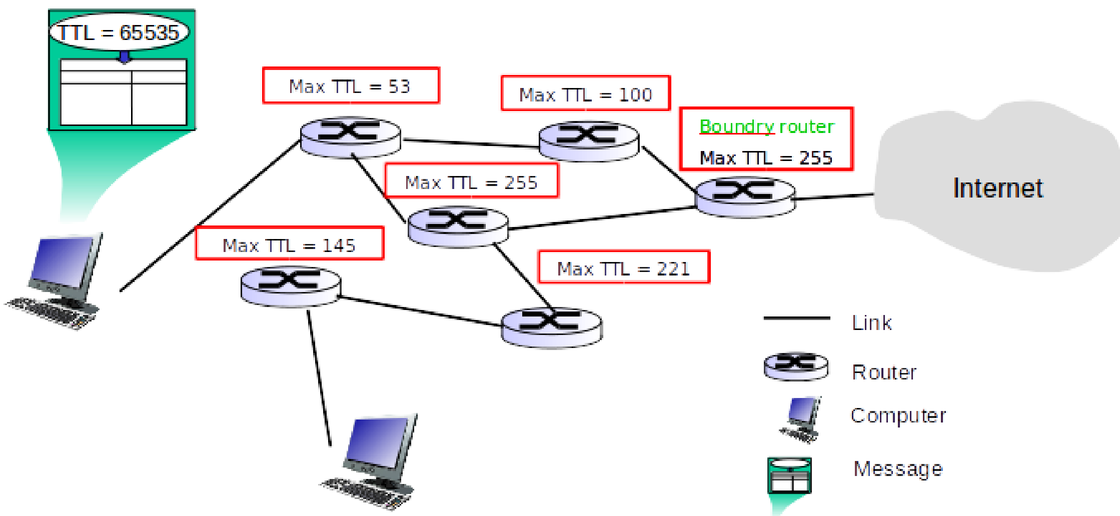
## Sample Output

```
5
9
```

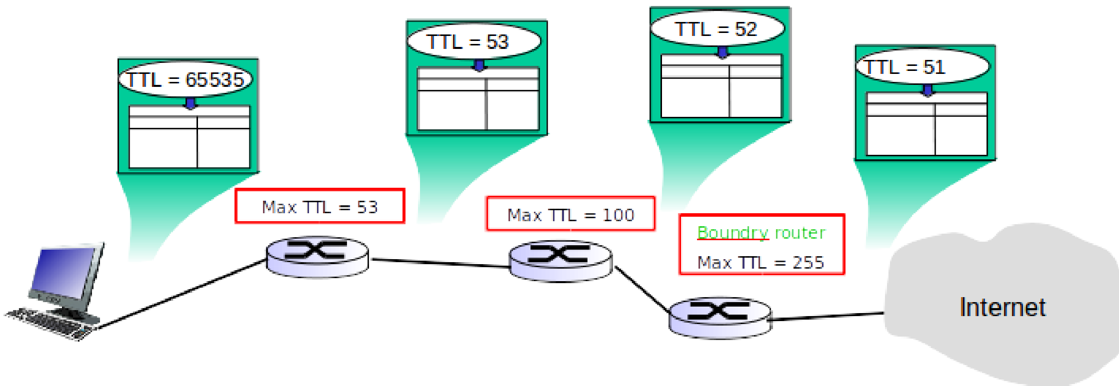
# C– Bahman's Network

## Description

In country of Utopia, Bahman has recently accepted the important task of constructing the nation's first network infrastructure. After months of research and analysis, he has realized that people will use their personal computers to connect to this network. Clients will have network connectivity as soon as they establish a link with one of the routers in the network. This initial router will be used to introduce their messages, i.e. packets will be first transmitted to this router (from client's machine) and from there will travel through series of intermediary routers to reach their destination. For destinations out of Utopia, messages should be first routed to certain boundary routers. These routers are directly connected to internet so they are capable of forwarding incoming packets to the global network. The example below depicts a network based on Bahman's design:




In this so-called network, each message has an attribute called TTL which holds maximum number of intermediary routers that message is allowed to go through. Messages will be created with an initial TTL value of 65535. Additionally, each router is assigned a TTL limit used to restrict the traffic passing through. Upon arriving at each router, message TTL value will be reduced by 1 and then the minimum of the new TTL value and the router's TTL limit will be assigned to the message. For example, consider a message $M_1$ with TTL value 65535 and a router $R_1$ with TTL limit 100. Upon arrival at $R_1$, $M_1$'s TTL value will be updated to 100 (minimum of 65534 and 100).




Bahman has little knowledge of rules governing the global network infrastructures, so to be on the safe side, he is expecting all outgoing messages to have the maximum possible TTL as they reach one of the boundary routers. Finally, he wants to know the minimum TTL value of messages reaching the network boundaries. In case a particular message can not reach the boundary, it's TTL value will be considered 0 in our calculations.

## Input

The input contains several test cases.
In the first line of input comes $T$ ($0 < T \leq 100$), the number of test cases.
For each test case, two positive integers $N$ and $M$ are provided which denote the number of routers and links respectively ($1 \leq N \leq 500$ and $1 \leq M \leq 3000$). On the next line, $N$ space-separated TTL values will be provided.
Next, we will have $n$ lines each describing a router by first providing the number of adjacent routers and then providing their identifiers. Note that routers are assigned $1...N$ as their identifiers. Finally, on the last line of input a parameter $k$ denoting the number of boundary routers and $k$ space-separated identifiers are given.

## Output

For each test case, print out the minimum value of messages as they reach network boundaries.

## Sample Input

```
3
2 1
100 100
1 2
1 1
1 2
4 4
100 5 10 5
2 2 3
2 1 3
3 1 2 4
1 3
2 2 4
3 1
100 100 100
1 2
1 1
0
1 3
```

## Sample Output

```
99
5
0
```

# D– 99 Red Balloons

## Description

Pejman is organizing the Internet ICPC contest of the University of Tehran. He needs a lot of balloons and asks Mohammad (the accountant of the scientific student chapter) for money. Mohammad has some problems with calculating the remaining money because of massive money transactions. The transactions are divided into three types as follows:

- Electronic transactions: The money that is added to/withdrawn from account via the internet bank gateway
- Strongbox transactions: The money that is added to/removed from strongbox placed in the chapter's office
- Future transactions: The money that will be paid/received at specific times in future

Mohammad has some bank accounts and all of bank transactions are Electronic transactions. Transactions between two different banks cost $p$ percentage commission which is paid from the source account, but there is no commission for internal transactions, the transactions between the accounts in the same bank. Banks are distinguished by the first two digits of the account numbers. Also, banks pay interest to their customers. The interest is calculated hourly but it is not added to the account instantly. The interests are accumulated and added to the account on the first day of the next month (when the time is 00:00). When an hour passes, $b$ percent of the current money is added to the interest. For example, if the account number 618546 contains 100$ at 14:25 of July 24th, 2014 and the value of $b$ is 0.5, the interest amount at 16:00 of the same day is 50 cents and at 17:00 is 1$. The calculations' accuracy of the banks is at most 0.00000001 dollar.
Strongbox transactions don't have any commission or interest. The incomming money (anything paid to Mohammad) is directly added to the strongbox. For all the spendings, first it is paid from the strongbox and if there is not enough money, Mohammad will use his accounts in the increasing order of account numbers to pay the remaining amount.
Mohammad has written all of transactions and he wants to calculate the amount of his money when Pejman wants to buy the balloons. There is a possibility that he cannot pay future payments without considering Pejman's balloons. Help Mohammad calculate the remaining money according to the given transactions.

## Input

The input contains several test cases.
In the first line of input comes $T$ ($0 < T \leq 30$), the number of test cases.
For each test case, in the first line there are 3 dates and times. The first date-time is when accounts have been opened, the middle one is the current date and time and the last one is when Pejman wants to buy the balloons. Dates and times are displayed as `yy/mm/dd-hh:mm`. Note that the number of days in each month is 31,28,31,30,31,30,31,31,30,31,30 and 31, starting from January. We consider only the years after

2000 and before 2100 (00 to 99). And we don't have any leap years!
In the second line, there are three integers: $s$, $n$ and $m$, where $0 \leq s \leq 1000, 0 < n \leq 10$ and $0 < m \leq 200$ are the initial amount of money, accounts count and transactions count, respectively.
In the third line, there are two non-negative real numbers: $p$ (a real number less than 20) which represents the commission percentage, and $b$ (a real number less than 0.1) which represents the interest percentage of banks.
In the next $n$ lines, there are two integers in each line: $bn$ (bank account number) and $a$ (initial amount of that account), where $bn$ is a 6-digit number and $0 \leq a \leq 1000$.
In the next $m$ lines, there is a transaction in each line as follows.

- Electronic transaction's format is "*%SBA %DBA %date-time %amount*". *%SBA* is the source bank account number, *%DBA* is the destination bank account number, *%date-time* is the date and time when the transaction occurred (before the current time) and *%amount*(dollar) is a positive integer.
- Strongbox transaction's format is "Safe *%date-time %amount*". *%date-time* is the date and time when the transaction occurred (before the current time) and *%amount* is an integer. If positive, it represents the amount of dollars that has been added to strongbox, or the amount of dollars that has been paid from it, otherwise.
- Future transaction's format is "*%BA %date-time %amount*". If *%amount* is positive, it will be added to strongbox; otherwise, added to *%BA* (the destination bank account number) at *%date-time* (it's in the future). Exchanging money from strongbox to a bank account doesn't have any commission.

In the last line, there are two numbers. The first one is a positive integer less than 500 that shows balloons' price and the second one is Pejman's bank account number.

## Output

For each test case,

- if there is no problem with future transactions and there is enough money to buy the balloons, output "`Happy Mohammad`".
- else if there is no problem with future transactions but Mohammad can't afford balloon's money, output "`Sad Mohammad`".
- otherwise (if there is some problem with future transactions and Mohammad will be in debt), output "`Sad Sad Mohammad`".

Pay attention to the order of outputs.

## Sample Input

```
2
13/05/14-09:16 14/10/27-18:25 14/11/12-12:00
500 2 6
1.5 0.002346
516482 100
493341 950
516482 502229 13/08/01-13:15 50
Safe 13/12/29-10:58 -300
516482 514832 13/11/07-15:01 50
493341 516482 14/09/20-11:31 800
631102 14/10/28-12:00 250
518240 14/11/03-10:30 500
150 512580
00/01/01-00:00 98/05/15-12:00 99/12/31-23:59
100 3 3
1.8 0.000023408
516482 500
493142 500
123456 500
516482 493142 05/05/05-05:05 496
493142 516482 06/05/05-05:05 980
526421 99/06/12-12:00 -1500
380 125268
```

## Sample Output

```
Happy Mohammad
Sad Mohammad
```

# E– **EdoceRahs**

## Description

"Edoce" and "Rahs" are twin sisters and they are very close to each other as often their friends call them "edocerahs". They have a special ability! They can manipulate numbers in their minds. Although they are twins, but they have different approaches and preferences in order to manipulate numbers. Edoce prefers odd numbers rather than even numbers and Rahs likes even numbers more than odd numbers. Their hobby is to choose any random number $N$ and write down the numbers 1 through $N$ on the paper. Say for $N = 12$ they write `123456789101112`. Edoce starts with eliminating all the digits in odd indices to create another sequence of numbers and then eliminate all the digits in even indices and so on. When only one digit remains, that is going to be $N^{th}$ Edoce's number. Rahs also does the same, but she starts with eliminating even indices first and then odd indices and so on (indices start from 0).
For $N = 12$, Edoce's first sequence would be: `13579012`. Then by eliminating even indices: `3702`, then odd indices: `30`, and then by removing the digit `3`, the remaining digit would be `0`, so the $12^{th}$ Edoce's number is `0`.
Rahs, on the other hand, first would have: `2468111`. Then by eliminating odd indexes: `2611`, then even indexes: `61`, and finally removing `1` and the remaining digit would be `6`, so the $12^{th}$ Rahs' number is `6`.

Write a program to compute $N^{th}$ Edoce's number and Rahs' number.

## Input

The input contains several test cases.
In the first line of input comes $T$ $(0 < T \leq 100001)$, the number of test cases.
Each line of the following $T$ lines is a case, containing the integer $N(0 \leq N < 100000)$.

## Output

For each case, write "`Edoce:X, Rahs:Y`" in a line which $X$ is $N^{th}$ Edoce's number and $Y$ is $N^{th}$ Rahs' number.

## Sample Input

```
2
4
12
```

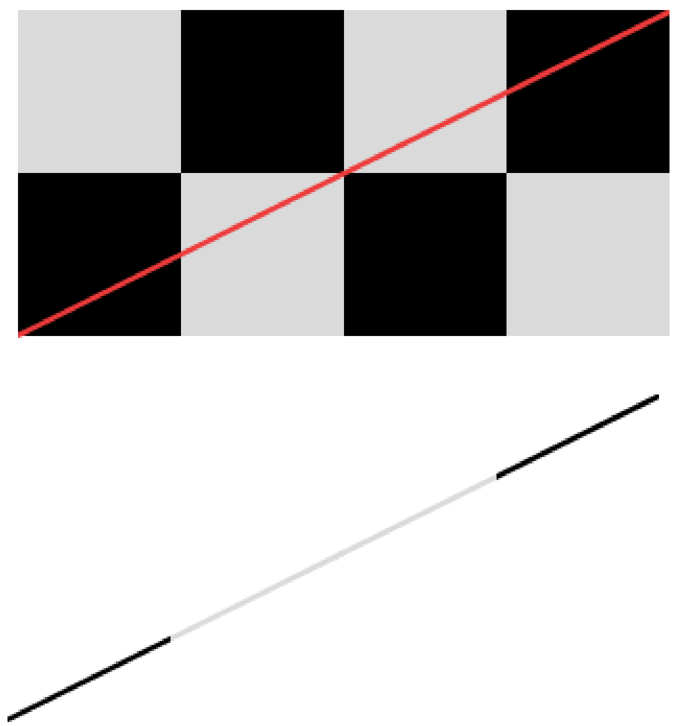
## Sample Output

```
Edoce:3, Rahs:2
Edoce:0, Rahs:6
```

# F– Tape the Art

## Description

Bahman is a perfect artist who produced some paintings in the shape of a chessboard. His paintings are all the same but different in width and height. He always chooses a piece of paper sized $Mcm \times Ncm$. He divides the paper with imaginary lines to create $M \times N$ little $1cm \times 1cm$ squares, and then he paints the squares with two colors (black and white) like a chessboard. In order to put his work in a gallery, they have to tape them into the wall. They will use a very low-width tape (just like a direct line) and will tape that from the top-right corner of the artwork to its bottom-left. As the tape should be colored exactly the same as the color of the square underneath, we want to compute the total length of the black part of the tape.



Assume the bottom-left square is always black.

## Input

The input contains several test cases.
In the first line of input comes $T$ $(0 < T < 10000)$, the number of test cases.
Each case is a line containing two integers $M$, $N (1 \leq M, N \leq 2^{31} - 1)$.

## Output

For each case, output a single line containing the length of the line colored black, rounded up to 3 digits after the decimal point.

## Sample Input

```
2
2 4
3 6
```

## Sample Output

```
2.236
3.354
```

# G– Khayyam - Pascal's Triangle

## Description

> "Pascal's triangle is a triangular array of the binomial coefficients. The rows of Pascal's triangle are conventionally enumerated starting with row $n = 1$ at the top. The entries in each row are numbered from the left beginning with $k = 1$. A simple construction of the triangle proceeds in the following manner. On row 1, write only the number 1. Then, to construct the elements of following rows, add the number above and to the left with the number above and to the right to find the new value. If either the number to the right or left is not present, substitute a zero in its place. For example, the first number in the first row is 1 (the sum of 0 and 1), whereas the numbers 1 and 3 in the fourth row are added to produce the number 4 in the fifth row."
>
> –*Wikipedia*

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1

We choose an arbitrary row like $R$ that its elements are $\{r_1, ..., r_R\}$ and we change this row in the following manner:

$$\forall i, i \text{ is odd} \land i < R \Rightarrow r_i = r_i + r_{i+1}$$

Now we want to know what happens to the $M^{th}$ entry in the $N^{th}$ row.
For example, if we choose the $2^{nd}$ row, then its elements change to $\{2, 1\}$ and then the $2^{nd}$ entry in the $4^{th}$ row would be 5.

## Input

The input contains several test cases.
In the first line of input comes $T$ $(0 < T < 64)$, the number of test cases.
Each test case consists of two lines. The first line contains an integer $R$ $(1 \leq R \leq 50000)$, the index of the row that we are going to change as described above.
The second line contains integers $N$ and $M$ $(R < N < 200000, 1 \leq M \leq N)$.

## Output

For each test case, output the $M^{th}$ element of the $N^{th}$ row of the triangle, modulo $1,000,000,009$ ($10^9 + 9$ is a prime number).

## Sample Input

```
2
5
7 5
7
9 5
```

## Sample Output

```
19
96
```

# H– IGPC

## Description

As more people are participating in IGPC (International Gossipful Programming Contest) and more people are watching the results, cheating in the contest has become very hard these days. But as we all know, Bahman is a very good cheater and all these years, he has managed to fool every single person involved in IGPC.
Bahman is the network administrator of the place where contest will be held. So it is easy for him to cheat in the contest as all the submissions will go throgh the network. But couple of weeks ago, a brave detective showed up and released some evidence which indicated that someone is cheating in the IGPC!
Luckily, Bahman knows exactly who this detective is and he knows all of his friends and his personality. So he can map the activities of the detective's brain and try to cheat in a way that the detective doesn't notice.
For the final round of these contests, it has been decided that the certificates be issued only for high rank teams. Teams with more than 3 solved problems will be counted as high rank. So it came to Bahman's mind that he can cheat and change the results of the teams so one team doesn't get the certificate and another team does.
The detective has some predictions. If any of those predictions doesn't come true, he immediately publishes more convincing evidence about cheating in IGPC. His predictions are in the form of one of these sentences:

```
At least one of TeamX or TeamY should get a high rank.
At least one of TeamX or TeamY should get a low rank.
TeamX and TeamY should get the same rank.
TeamX and TeamY should not get the same rank.
Both TeamX and TeamY should get a high rank.
Both TeamX and TeamY should get a low rank.
```

On the other hand, Bahman receives lots of requests from teams and their coaches and even the observers who want to cheat in the contest. Requests are in the form of one of these sentences:

```
I will pay you D dollars if TeamX and TeamY get a high rank.
I will pay you D dollars if TeamX and TeamY get a low rank.
I will pay you D dollars if TeamX gets a high rank and TeamY gets a low rank.
I will pay you D dollars if TeamX gets a low rank and TeamY gets a high rank.
```

Now, as you may already know, Bahman wants to cheat and change the result of some submissions from "accpeted" to "wrong answer" or from "wrong answer" to "accepted", to give some teams a high or low rank. He may also submit a code on behalf of a team. But according to the requests, he wants to get as much money as he can and at the same time, he should watch out and be careful not to break any of the detective's predictions.

## Input

The input contains several test cases.
In the first line of input comes $T$ ($0 < T \leq 25$), the number of test cases.
In the first line of each test, there is an integer $n$ ($0 \leq n \leq 100$), the number of teams. In the next line there is an integer $r$ ($1 \leq r \leq 5000$) which is the number of cheating requests. Each of the next $r$ lines contains the description of a cheating request as described above ($0 \leq D < 64$). The next line contains the integer $p$ ($0 \leq p \leq 100$), the number of predictions. In the following $p$ lines, there will be one prediction in each line as noted earlier.
Team names contain only English alphabets and numbers, are case-sensitive and at most 15 characters in length.

## Output

For each test case, output the maximum profit that Bahman can get by manipulating the results, while not breaking any of the detective's predictions.
If there is no possible way to not break the predictions, just print "`Haters gonna hate anyway!`" in one line.

## Sample Input

```
2
3
2
I will pay you 9 dollars if BMP gets a high rank and HereForPunch gets a low rank.
I will pay you 7 dollars if MASSters and MASSters get a high rank.
2
HereForPunch and MASSters should not get the same rank.
At least one of BMP or MASSters should get a low rank.
10
1
I will pay you 50 dollars if HereForLunch and ZoMbeeZ get a high rank.
2
Both HereForLunch and ZoMbeeZ should get a high rank.
Both 0x00 and HereForLunch should get a low rank.
```

## Sample Output

```
7
Haters gonna hate anyway!
```

# I– Obsessive Behavior

## Description

Bahman is a disciplined person when it comes to decisions. Even decisions like making new friends! At school, he always made a list of people, whom he found cool, according to how much he liked them. He has his lists for primary school ("Dabestaan" in Persian), secondary school ("Raahnamaaei" in Persian) and high school ("Dabirestaan" in Persian) now.
Bahman is going to enter university, but he wants to make the decision of choosing a university based on his friends.
Specifically, he knows that if he liked his friend $X$ more than $Y$ in one of his lists ($X$ comes before $Y$ in the list), then he would like $X$ more than $Y$ in university (if they all go to the same university).
Unfortunately, Bahman's friends have chosen their universities and Bahman knows this. So he wants to choose a university where he can get the longest list of friends in the same order he made his previous lists. This list should only contain the friends that went to the same school with Bahman from primary school to high school.
As you might have noticed, despite the confusion, it is not odd to have friends with the same name. So you can safely assume that if Bahman had a friend called $X$ who is going to university $U$ in both primary school and high school, they are the same person.
Given Bahman's three lists and his friends' chosen universities, help him find the best university and the list of his old friends there.
Dont worry! He would definitely go to a university!

## Input

The input contains several test cases.
In the first line of input comes $T$ $(0 < T < 100)$, the number of test cases.
Each test case has 9 lines which describe 3 lists (3 lines for each list) in the order of primary school, secondary school and high school.
The first line for each list would be $n$ $(1 \leq n \leq 300)$, the number of friends in that list. The second line contains $n$ case sensitive strings, each of which is at most 100-English-alphabet-characters long. The $i^{th}$ string is the name of the $i^{th}$ person on the list. The third line contains $n$ integers between 1 and 10 inclusive. The $i^{th}$ integer is the chosen university by the $i^{th}$ person.

## Output

For each test case, print two lines. The first line should contain the number of the university Bahman should choose, and the second line should contain the names of friends in the correct order, separated with a space.
If there exists more than one possible answer, print any of them.
If the list is empty, print a blank line.

## Sample Input

```
3
4
A A A A
1 2 2 1
4
A A A A
2 1 2 1
4
A A A A
1 2 1 2
3
A B C
1 2 3
3
B C A
1 2 3
3
C A B
1 2 3
9
Mohsen Aryaz Ali Pejman Mohammad Mohammad Ali Sina Mohammad
2 2 1 2 2 1 1 3 2
8
Mohsen Ali Pejman Aryaz Mohammad Mohammad Hassan Mohammad
2 1 2 2 2 1 4 2
9
Mohsen Hassan Ali Sina Pejman Mohammad Sina Aryaz Mohammad
2 4 1 3 2 2 1 2 2
```

## Sample Output

```
2
A A
1

2
Mohsen Pejman Mohammad Mohammad
```

# J– Solar Power

## Description

Today, the increased consumption of energy in industrial societies has brought about irreversible environmental changes and renewable energy sources like solar power can play an important role toward eliminating these threats. The school of Electrical and Computer Engineering at University of Tehran, one of the pioneers in the field of renewable energies, has recently opened a new solar complex on campus. Their revolutionary solar panels will harness radiant light and heat from the sun and transform it to electrical energy. In addition to supplying the university infrastructure, they are planning to sell the excess amount of electricity to residential areas on the opposite side of North Kargar Street.
Tehran municipal authorities has provided several electrical cables from one side of the street to the other side.
Unfortunately, some of these cables cross each other and it is safe to say that the current passing through will cause serious damages to the network. Hence for each pair of crossing cables, we are only allowed to choose one.
In addition, each of these cables has a limit on how much current ($Amperes/Hour$) it can let through without risking the network's integrity. UT technicians are tasked with selecting a subset of these cables that maximizes the current in terms of $A/H$. Write a program to help them achieve this goal.

## Input

The input contains several test cases.
In the first line of input comes $T$ ($0 < T \leq 100$), the number of test cases.
For each test case, in the first line three positive integers $m$, $n$, and $k$ are given which respectively denote the number of electrical pylons in the right side of street, the number of electrical pylons in the left side, and the total number of provided cables ($m, n \leq 1000$ and $k \leq mn$). Each of the following $k$ lines contains three positive integers ($i$, $j$, and $w$, $1 \leq i \leq m$, $1 \leq j \leq n$) and one string ($id$) which represent the information of each cable; there is a cable with the name of $id$ from electrical pylon $i$ in the right side to the electrical pylon $j$ of the left side with the maximum amount $w$ Ampere per hour. There is at most one cable between each two nodes. Ids are unique strings containing only the lower case English alphabets with maximum length of 64.

## Output

For each test case, print two lines. The first line contains the maximum amount of $A/h$ and the second one ids of selected cables which are separated with space. It should be noted that the ids should be printed in ascending alphabetical order.
If there is more than one possible solution, output any of them.

## Sample Input

```
2
3 4 5
1 2 1 a
2 1 2 b
3 4 1 c
3 3 2 d
2 4 1 e
2 2 4
1 1 1 a
1 2 2 aa
2 1 3 aaa
2 2 4 aaaa
```

## Sample Output

```
5
b c d
8
a aaa aaaa
```

# K– **Free Lunch**

## Description

Bahman is participating in a very popular cooking game show. The prize of the winner is a free lunch at the ACM ICPC.
The goal of the game is to make the best cake out of some raw cooking ingredients. The judges are very picky, so Bahman needs to follow the instructions of making the cake better very precise. The host provided the participants all the $N$ raw ingredients they can use. All these $N$ ingredients are in separated containers, ordered $C_1$ to $C_n$. The only rule of the game is that you can not take ingredients freely. This rule is as follows.

1. You choose any container $C_i$ and any positive integer $d$.
2. You take exactly $d$ grams of ingredient from container $C_i$.
3. You choose $K > 0$ containers from the set $\{C_j \mid j < i\}$, and get some ingredients (in grams, integer) from them, in a way that the total weight of those ingredients from these $K$ containers becomes equal to $d$.

You can repeat this procedure as many times as needed, but you can not throw away any grams of ingredients you take.
Bahman has several cooking recipes, but he does not know if it is possible to take the required ingredients by the rule of the game or not. Assume that the capacity of each container is infinite.
Write a program that given the required ingredients for a recipe, determines the possibility of taking that ingredients by the rule.

## Input

The input contains several test cases.
In the first line of input comes $T$ $(0 < T < 64)$, the number of test cases.
For each case, the first line contains an integer $N(1 \leq N \leq 10^6)$ indicating the number of containers. Then followed by $N$ lines, the $i^{th}$ line contains an integer $D_i(0 \leq D_i \leq 10^9)$ which shows we need $D_i$ grams of the $i^{th}$ container's ingredient.
**Caution!** Use fast IO functions as the input file is very big.

## Output

For each test case, output a line with the string "`YAAAYYYY`" if Bahman can win the game and "`NOOOO`" otherwise.

## Sample Input

```
2
3
2
4
2
3
1
2
2
```

## Sample Output

```
YAAAYYYY
NOOOO
```

# L– **Perfect Address**

## Description

Bahman, a distinguished employee of local post office, is responsible for designating postal addresses to residences located in North Kargar Street. Each postal address is a sequence of letters placed together without any white spaces in between. Furthermore, Bahman has a specific collection of keywords which he will use as he'll come up with new addresses.
Residences in this street are labeled $A_1$ through $A_n$ and $A_i$'s address will be calculated based on $A_{i-1}$'s address in the following manner:

- Bahman can choose a keyword from his collection and append it to $A_{i-1}$'s address.
- Do as above but remove the duplicate letters in case the chosen keyword and $A_{i-1}$'s address overlap, i.e. keyword's first $x$ letters are the same as address's last $x$ letters.

For example, consider $A_1$'s address to be '`abc`' and Bahman using '`bcdef`' as keyword. In this case, $A_2$'s address can either be '`abcbcdef`' or '`abcdef`'.
Bahman already knows that his boss, Robin, is planning to buy a house in this street and he prefers to have a postal address not bigger than $L$ letters. In addition, Robin has certain favorite keywords and his satisfaction of the address is directly influenced by how many times these keywords will appear in the final address.
Bahman is desperate to fully satisfy Robin by arranging the address to have as many favorite keywords as possible.
For example, consider Bahman holding keywords {`ab,ba`}, Robin favoring keywords {`aa,ab`}, and $L$ is equal to 4. The Collection of available addresses is listed below:

| Address | Robin's Satisfaction |
|---|---|
| `ab` | 1 |
| `ba` | 0 |
| `aba` | 1 |
| `bab` | 1 |
| `abab` | 2 |
| `abba` | 1 |
| `baab` | 2 |
| `baba` | 1 |

Your task is writing a program to find Robin's satisfaction for the perfect address.

## Input

The input contains several test cases.
In the first line of input comes $T$ ($0 < T < 64$), the number of test cases.
For each test case, on the first line it will come the integer $N$ ($1 \leq N \leq 50$) and the next $N$ lines hold Bahman's keywords, one in each line. After that, the integer $M$ will be given and the following $M$ lines will have Robin's keywords, one per line. The last line contains the integer $L$ ($1 \leq L \leq 50$) which denotes the maximum length of the desired address.
Keywords in both lists will have a maximum length of 10 and conatin only lower-case English characters.

## Output

For each test case, print a line with an integer which is the maximum satisfaction of Robin.

## Sample Input

```
2
2
ab
ba
2
aa
ab
4
1
i
1
iii
10
```

## Sample Output

```
2
8
```