Problem A
Fast Food Prizes
Problem ID: fastfood
Time Limit: 2 seconds

Around regional contest time, the Canadian branch of a popular fast food restaurant usually runs a game to promote its business. Certain food items provide stickers, and certain collection of different stickers can be converted to cash prizes. If a prize requires sticker types $T_1, T_2, \ldots, T_k$, then you can claim the prize if you have 1 sticker of each type $T_1, T_2, \ldots, T_k$. Each sticker can only be used to claim one prize. However, you may claim a prize multiple times if you have multiple stickers of the same type. No two prizes will require the same type of stickers. There may be some stickers that cannot be used to claim a cash prize (e.g. a sticker for a free milkshake).

On your road trip to the regional contest, your coach forced you to eat at this restaurant and collected all the stickers together. How much cash can your coach claim?

Input

The input consists of multiple test cases. The first line of input is a single integer, not more than 1000, indicating the number of test cases to follow. Each case starts with a line containing two integers n ($1 \leq n \leq 10$) and m ($1 \leq m \leq 30$), where n is the number of different types of prizes, and m is the number of different types of stickers (the types are labelled $1, 2, \ldots, m$). The next n lines specify the prizes. Each of these lines starts with an integer k ($1 \leq k \leq m$) specifying the number of sticker types required to claim the prize. This is followed by k integers specifying the types of the stickers required. The final integer on each line is the (positive) cash value of the prize (at most 1,000,000). The last line of each case gives m nonnegative integers, with the ith integer giving the number of stickers of type i your coach has collected. There are no more than 100 stickers of each type.

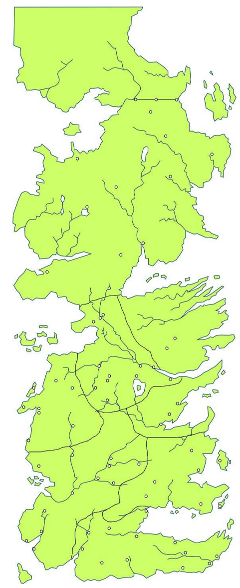
Output

For each case, display on a single line the total value of the cash prizes that can be claimed.

Sample Input	Sample Output
3 2 10 3 1 2 3 100 4 4 5 6 7 200 2 3 1 4 5 2 2 1 3 4 3 6 2 1 2 100 3 3 4 5 200 1 6 300 1 2 3 4 5 6 3 6 2 1 2 100 3 3 4 5 200 1 6 300 1 2 0 4 5 6	500 2500 1900

Problem B
Seven Kingdoms
Problem ID: kingdom
Time Limit: 9 seconds

Jon Dayne is the ruler of a huge country called Seven Kingdoms. He has two sisters, Arya and Sansa, and wants to give some cities of Seven Kingdoms to them. He will rule the remaining cities or if no city remains, goes to the Wall, a colossal fortification along the northern border of the Seven Kingdoms, to be the Lord commander. Arya is the Lady of Winterfell and Sansa is the Lady of King's Landing. The cities in Seven Kingdoms (including Winterfell and King's Landing) are connected to each other with a network of roads (although some cities may be disconnected from the other cities, because they are either located on an island or they are currently at war with these other cities). There is no direct road between Winterfell and King's Landing and they do not share a common neighbour city.



Picture under GNU Free Documentation License

Jon wants to assign a collection of cities to each one of his sisters such that each city in a collection is connected with a direct road to all other cities in that collection and the remaining cities, not in these two collections, are also connected with a direct road to each other. The collection assigned to Arya must include Winterfell and the collection assigned to Sansa must include King's Landing. Jon needs your help to determine whether this is possible and if this is possible, you should tell him the cities in each collection.

Input

The input consists of a single test case. The first line contains two integers n and m, where n ($2 \leq n \leq 2000$) is the number of cities, m is the number of roads. Each of the next m lines contains two integers x_i and y_i ($1 \leq x_i, y_i \leq n$) describing one road, where x_i and y_i are the distinct cities the road connects. Winterfell is city 1 and King's Landing is city 2 in the road network.

Output

If it is not possible to partition the cities in the way explained, display the word `impossible`. Otherwise, display two lines: the first containing the cities in the collection assigned to Arya and the second containing the collection of cities assigned to Sansa. If there are many such collections, any one of them is acceptable.

Sample Input	Sample Output
9 11 1 4 5 4 1 5 6 2 6 7 7 2 3 8 3 9 8 9 6 8 5 9	1 4 5 2 6 7

Sample Input	Sample Output
9 11 1 4 5 4 1 5 6 2 6 7 7 2 3 8 3 9 3 8 6 8 5 9	impossible

Problem C
Institute of Advanced Category Manipulation
Problem ID: acm
Time Limit: 1 second

A certain magazine annually ranks universities across the country in various categories, as a service for readers who may be deciding which university to attend. Each university is given three scores R, T, and S (integers from 0 to 100) for its research, teaching, and community service, respectively. These scores are then interpreted somehow and the top universities in a number of categories are named. There are many categories to compete in (research university, undergraduate university, small university, etc.). Sometimes these categories appear to be invented just to make a particular university a "top university" (e.g. top university in rural area with population 30,000 to 45,000).

The Institute of Advanced Category Manipulation (ACM) is not pleased with its ranking, because it is not a top university in any category. The public relations department of ACM has decided that it can invent its own category of universities, and rank each university based on a linear combination of the three scores. The combined score U for each university is calculated by the formula:

$$U = aR + bT + cS$$

where a, b, c are constant real numbers. To make sure that the readers do not notice the scores suddenly getting too large or too small, it is required that $a^2 + b^2 + c^2 = 1$. Since some of the scores for ACM are low compared to its peers, they will even allow any of a, b, and c to be negative. A university is considered a top university in the new category if its combined score is greater than or equal to the combined scores of each of the other universities. Each category is defined by the constants (a, b, c) satisfying the constraints given above.

This idea is quickly catching on among university administrators. For example, the Research University of Ivory Towers has decided to use $(a, b, c) = (0.57735, -0.57735, -0.57735)$. Even with this manipulation, some universities cannot be considered top universities. In the second case of the sample input below, The University of Mediocrity has a score of $R = T = S = 50$. It cannot be considered a top university among the given universities regardless of how it manipulates the categories.

In this problem, you will be given the research, teaching, and community service scores of some universities. Your task is to determine which of these universities can be considered a top university.

Input

The input consists of multiple test cases. The first line of input is a single integer, not more than 10, indicating the number of test cases to follow. The first line of each case is a positive integer n $(1 \leq n \leq 50)$ specifying the number of universities to be considered. The following n lines each contains three integers specifying the research, teaching, and community service scores.

Output

For each case, display on a single line a string consisting of n characters. The ith character is `T` if the ith university given in the input can be considered a top university and `F` otherwise.

Sample Input

```
2
4
100 0 0
0 100 0
0 0 100
100 100 100
9
100 0 0
0 100 0
0 0 100
100 100 0
100 0 100
0 100 100
100 100 100
0 0 0
50 50 50
```

Sample Output

```
TTTT
TTTTTTTTF
```

Problem D
Digital Content Protection
Problem ID: content
Time Limit: 2 seconds

Picture from Wikimedia Commons

Dan is working for a digital content protection company, which is responsible for the content protection of blu-ray discs based on a standard called Anti Content Misuse (ACM).

The ACM standard works as follows. Assume there are 2^n blu-ray drives/players. We represent these 2^n drives as the leaves of a complete binary tree of height n, so that each root-to-leaf path consists of n edges. Each node u in this binary tree is assigned an identifier number and contains a random key k_u. The identifier numbers are assigned as follows. The root, r, is assigned 1. In addition, the left and right children of an internal node having number i are assigned numbers $2i$ and $2i + 1$, respectively. This scheme assigns a distinct number to each node in the tree. The keys contained in the nodes are unknown to blu-ray users, but they are available to blu-ray drive manufacturers. Each blu-ray player is assigned the identifier number i ($2^n \leq i \leq 2^{n+1} - 1$) of its corresponding leaf in the tree. A manufacturer of blu-ray drives embeds the keys associated with the nodes in the path from the root to leaf number i in player number i.

To encrypt the content of a blu-ray disc, the company in charge creates a random key k called the master key. First, they encrypt k with the key k_r (recall r is the root node of binary tree) and write it on the disc as a header. Then, they encrypt the content with k, and write the encrypted data on the blu-ray disc. A blu-ray drive first decrypts the header using key k_r embedded in it and recovers the master key k and then, decrypts the content using the key k.

Unfortunately, the keys embedded in a set of blu-ray drives, R, are exposed by hackers and published on the web. As a result, we cannot encrypt the master key k using any of these exposed keys. For example, since all blu-ray drives contain k_r, the encryption scheme above does not work any more. There is a solution oversaw for this situation in the ACM standard. At the cost of a larger header, the industry can safely encrypt the content of a new blu-ray disc. They carefully choose a subset of unexposed keys K in the binary tree such that all blu-ray drives, except for drives in R, have at least one of the keys in K. They encrypt the master key k with each key $k' \in K$ and put the result in the header (i.e., there are $|K|$ ciphertexts in the header). Now, each active blu-ray drive can decrypt at least one of the ciphertexts in the header and can recover the master key k. Dan needs your help to determine a subset of keys K with minimum cardinality (which results in the smallest header) given the identifiers of hacked drives.

Input

The input consists of a single test case. A test case consists of two lines. The first line contains two integers n and $|R|$, where $1 \leq n \leq 62$ and $1 \leq |R| \leq 1000$. $|R|$ is the cardinality of R, the set of exposed drives. The second line contains $|R|$ integers, which are the identifiers of exposed blu-ray drives. You can assume that there is at least one blu-ray drive not hacked.

Output

Display the identifiers of nodes corresponding to the keys in K, satisfying the above requirements and having minimum cardinality, in increasing order and separated with single spaces.

Sample Input	Sample Output
`2 1` `5`	`3 4`

Sample Input	Sample Output
`3 3` `10 11 12`	`4 7 13`

Problem E
Credit Card Payment
Problem ID: creditcard
Time Limit: 2 seconds

Picture from Wikimedia Commons

Using credit cards for your purchases is convenient, but they have high interest rates if you do not pay your balance in full each month.

The interest rate is commonly quoted in terms of "annual percentage rate" (APR) which is then applied to the outstanding balance each month. The APR can be converted to a monthly interest rate R. At the end of each month, the monthly interest rate is applied to the outstanding balance and the interest is added to the total balance. Any payment made will be applied to the balance in the following month. The monthly interest is rounded to the nearest cent (rounding up 0.5 cent and above) in the calculations.

You have unfortunately accumulated an outstanding balance B at the end of the month and you can only afford to pay up to some amount M every month. If you do not make any more purchases with the credit card, what is the minimum number of payments needed to completely eliminate the outstanding balance? It is possible that you cannot pay off the balance in 100 years (1200 payments).

Input

The input consists of multiple test cases. The first line of input is a single integer, not more than 1000, indicating the number of test cases to follow. Each of the following lines specify the input for one case. Each line contains three positive real numbers separated by single spaces: R, B, and M. The real numbers have two digits after the decimal point, satisfying $R \leq 50.00$ and $B, M \leq 50000.00$. R is the monthly interest rate and is specified as a percentage.

Output

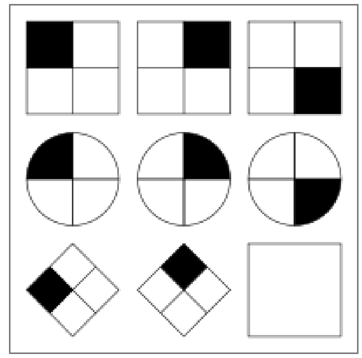
For each case, display on a line the minimum number of payments needed to eliminate the outstanding balance. If this cannot be done in at most 1200 payments, print instead `impossible`.

Sample Input	Sample Output
11 2.00 100.00 105.00 2.00 100.00 102.00 2.00 100.00 100.00 2.00 100.00 4.00 2.00 100.00 3.00 2.00 100.00 1.00 2.00 100.00 2.00 9.56 5462.50 522.22 12.50 29876.44 33610.99 5.50 1.00 1.05 14.78 40181.09 46119.86	1 1 2 36 56 impossible impossible impossible 2 2 1

Problem F
IQ Test
Problem ID: iq
Time Limit: 2 seconds

In many IQ tests, the following type of questions is often given:

> Given the first few terms of an integer sequence, what is the next term?



Picture from Wikimedia Commons

For example, if you are given the sequence 1, 1, 2, 3, 5, 8, 13, 21 you may recognize this as the Fibonacci numbers and write down 34 as the next term.

There is no "correct answer" because the next term can be any integer and still be generated by a polynomial (possibly of a very high degree). In this problem, we are only interested in sequences that satisfy a recurrence relation of the form

$$f(n) = a_1 f(n-1) + \dots a_d f(n-d),$$

where $1 \leq d \leq 3$, and $a_1, \dots, a_d$ are integers. If the sequence satisfies multiple recurrence relations of the type above, we will always prefer one with a smaller d.

Input

The input consists of multiple test cases. The first line of input is a single integer, not more than 500, indicating the number of test cases to follow. Each case is specified on one line. Each line contains a number of integers: the number of given terms in the sequence n ($8 \leq n \leq 12$), followed by n integers containing the given sequence. Each of the given terms has absolute values at most 1000. You may also assume that the given sequence satisfies at least one recurrence relation in the form described above. The first d terms in the given sequence are non-zero, for the smallest d for which a recurrence exists.

Output

For each case, display on a line the next term generated by the recurrence relation selected by the criteria above. You may assume that the next term in the sequence has absolute value at most 100,000.

Sample Input	Sample Output
3 8 1 1 2 3 5 8 13 21 8 1 1 1 1 1 1 1 1 8 1 -2 4 -8 16 -32 64 -128	34 1 256

This page is intentionally left blank.

Problem G
Social Advertising
Problem ID: social
Time Limit: 2 seconds

You have decided to start up a new social networking company. Other existing popular social networks already have billions of users, so the only way to compete with them is to include novel features no other networks have.

Your company has decided to market to advertisers a cheaper way to charge for advertisements (ads). The advertiser chooses which users' "wall" the ads would appear on, and only those ads are charged. When an ad is posted on a user's wall, all of his/her friends (and of course the user himself/herself) will see the ad. In this way, an advertiser only has to pay for a small number of ads to reach many more users.

You would like to post ads to a particular group of users with the minimum cost. You already have the "friends list" of each of these users, and you want to determine the smallest number of ads you have to post in order to reach every user in this group. In this social network, if A is a friend of B, then B is also a friend of A for any two users A and B.

Input

The input consists of multiple test cases. The first line of input is a single integer, not more than 10, indicating the number of test cases to follow. Each case starts with a line containing an integer n ($1 \leq n \leq 20$) indicating the number of users in the group. For the next n lines, the ith line contains the friend list of user i (users are labelled $1, \ldots, n$). Each line starts with an integer d ($0 \leq d < n$) followed by d labels of the friends. No user is a friend of himself/herself.

Output

For each case, display on a line the minimum number of ads needed to be placed in order for them to reach the entire group of users.

Sample Input	Sample Output
2 5 4 2 3 4 5 4 1 3 4 5 4 1 2 4 5 4 1 2 3 5 4 1 2 3 4 5 2 4 5 2 3 5 1 2 2 1 5 3 1 2 4	1 2

Problem H
Tables
Problem ID: tables
Time Limit: 2 seconds

HTML uses a simple tag format for table layout. You are to create ASCII-art tables based on a simplified notation.

Logically, a table can be considered an m by n grid, each with a 2-character wide by 1-character high cell, for example a 2×3 grid would be as follows:

```
 -- -- --
|11|12|13|
 -- -- --
|21|22|23|
 -- -- --
```

The output consists of $2m + 1$ rows, each having $3n + 1$ characters (including leading and trailing spaces in odd-numbered rows).

But some tables are not strictly grid-based, because the values for certain cells can span multiple rows and/or columns. Here is the layout if cell 11 has a row span of 2 and cell 22 has a column span of 2:

```
 -- -- --
|11|12|13|
    -- --
|  |22   |
 -- -- --
```

You are to create these ASCII-art tables given m, and the row span and column span values.

Input

The input consists of multiple test cases. The first line of each test case contains one integer m, the number of rows. Then, m lines follow, describing the row and column spans in the layout. Each of the next m lines gives the span information for a row. The span information is given only for the upper left cell in the span. If N row spans and column spans need to be specified for a given row, these would be specified as:

$$N\ \mathtt{RS}_1\ \mathtt{CS}_1\ \ldots\ \mathtt{RS}_N\ \mathtt{CS}_N$$

where the values $\mathtt{RS}_k$ and $\mathtt{CS}_k$ are between 1 and 9, and specify how many rows and columns the next cell needing information in this row needs to occupy. Previous row spans or column spans may imply that fewer cells than the total number of columns need to be specified for a particular row.

You may assume that the input is valid:

- the number of rows and columns are between 1 and 9;
- every row will have the same number of columns;

- there are no overlapping spans of cells;
- every cell is contained in some span of cells.

There are at most 100 test cases and the end of input is indicated by $m = 0$.

Output

For each case, display the ASCII-art table, with the row-column index displayed in the top left corner of every span of cells. Display a blank line after every case.

Sample Input

```
3
2 1 1 1 1
2 1 1 1 1
2 1 1 1 1
3
2 1 1 2 1
1 1 1
2 1 1 1 1
3
1 1 2
2 1 1 1 1
2 1 1 1 1
3
2 2 1 1 2
1 2 2
1 1 1
0
```

Sample Output

```
 -- --
|11|12|
 -- --
|21|22|
 -- --
|31|32|
 -- --

 -- --
|11|12|
 --
|21|  |
 -- --
|31|32|
 -- --

 -- --
|11   |
 -- --
|21|22|
 -- --
|31|32|
 -- --

 -- -- --
|11|12   |
    -- --
|  |22   |
 --
|31|     |
 -- -- --
```

Problem I
Flood-It
Problem ID: floodit
Time Limit: 2 seconds

Flood-It is a popular one player game on many smart phones. The player is given an $n \times n$ board of tiles where each tile is given one of 6 colours (numbered 1–6). Each tile is connected to up to 4 adjacent tiles in the North, South, East, and West directions. A tile is connected to the origin (the tile in the upper left corner) if it has the same colour as the origin and there is a path to the origin consisting only of tiles of this colour.

A player makes a move by choosing one of the 6 colours. After the choice is made, all tiles that are connected to the origin are changed to the chosen colour. The game proceeds until all tiles have the same colour. The goal of the game is to change all the tiles to the same colour, preferably with the fewest number of moves possible.

It has been proven that finding the optimal moves is a very hard problem. For this problem, you will simulate a very simple greedy strategy to see how well it works:

1. for each move, choose the colour that will result in the largest number of tiles connected to the origin;
2. if there is a tie, break ties by choosing the lowest numbered colour.

To illustrate this, we look at the first test case in the sample input, the original board is:

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

If we choose colour 3 for the first move, the result will be:

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

where the tiles connected to the origin are shaded. In the next move, we choose colour 4 because we can increase the number of tiles connected to the origin by 5 tiles:

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

Input

The input consists of multiple test cases. The first line of input is a single integer, not more than 20, indicating the number of test cases to follow. Each case starts with a line containing the integer n ($1 \leq n \leq 20$). The next n lines each contains n characters, giving the initial colours of the $n \times n$ board of tiles. Each colour is specified by a digit from 1 to 6.

Output

For each case, display two lines of output. The first line specifies the number of moves needed to change all the tiles to the same colour. The second line specifies 6 integers separated by a single space. The ith integer gives the number of times colour i is chosen as a move in the game.

Sample Input

```
4
6
123423
334521
433123
543621
324343
234156
5
12121
21212
12121
21212
12121
5
12345
12345
12345
12345
12345
5
11131
12211
31311
21111
11111
```

Sample Output

```
12
2 2 4 2 1 1
8
4 4 0 0 0 0
4
0 1 1 1 1 0
4
1 2 1 0 0 0
```

Problem J
Chemicals Monitoring
Problem ID: chemicals
Time Limit: 4 seconds

Picture from Wikimedia Commons

Victor works for Alberta Chemicals Monitoring (ACM). ACM is a company that analyses raw environmental data related to chemicals used in oil sand and other industries in Alberta, and produces some reports for environmental watchdogs.

Victor is in charge of a multi-processor cluster in ACM. Each processor is connected to a dedicated special purpose output generation unit (OGU). This cluster receives several raw data streams from field sensors and assigns each stream to a processor. Each processor performs some real time processing on a data stream and immediately after its termination, produces a report using its OGU.

Each stream has an integer starting time s, an integer duration d and a priority p. This stream is active in the interval $[s, s + d)$ (right-open interval). The report of each stream must be produced immediately after its termination; otherwise, it will be useless. An OGU creates a report extremely fast, so you can assume that an OGU produces this report instantly.

In the past, at any instance of time, the number of data streams were not more than the number of processors and OGUs. So, Victor could process all data streams. Unfortunately, recently, in a suspicious power surge, all OGUs burnt out. Victor was able to salvage one OGU by using parts from the other OGUs. Now, he can no longer produce a report for all data streams and needs to choose a subset of them based on the priorities assigned to them. To handle access to this OGU, Victor restructured the cluster architecture as follows. When a stream starts, the system either admits or rejects it. If it admits a stream, the unique identifier of the processor assigned to this stream is pushed onto the stack. Only a processor having its identifier on top of the stack can use the OGU to produce its report. After production of the report, the processor identifier is popped from the stack. It should be noted that if some streams start at the same time, he can push their processor identifier in any order of his choice. Now, Victor needs your help to choose a subset of streams such that their reports can be generated with this single OGU. The total priority of the streams in the chosen subset should be maximized.

Input

The input consists of a single test case. The first line contains an integer n, where n ($1 \leq n \leq 5000$) is the number of data streams. Each of the next n lines contains three integers s_i, d_i, p_i ($1 \leq s_i, d_i \leq 10^9$, $0 \leq p_i \leq 100,000$) describing one data stream, where s_i is its start time, d_i is the duration of the stream, and p_i is its priority. Note that the cluster has at least 5000 processors.

Output

Display the maximum total priority of a subset of streams such that their reports can be generated with the architecture described above using a single OGU.

Sample Input	Sample Output
4 1 3 6 2 5 8 3 3 5 5 3 6	13

Sample Input	Sample Output
6 5 4 10 3 4 6 1 8 100 3 2 3 4 2 4 3 2 2	115