11628 - Another lottery

Time limit: 1.000 seconds

Even in times of an economic crisis, people in Byteland still like to participate in lotteries. With a bit of luck, they might get rid of all their sorrows and become rich.

The most popular lottery in Byteland consists of m rounds. In each round, everyone can purchase as many tickets as he wishes, and among all tickets sold in this round, one ticket is chosen randomly, each one with the same probability. The owner of that ticket wins the prize money of this round. Since people in Byteland like powers of 2, the prize money for the winner of round i amounts to 2^i Bytelandian Dollars.

Can you determine for each participant in the lottery the probability that he will win more money than anybody else?

Input Specification

The input consists of several test cases. Each test case starts with a line containing two integers n and m, the number of participants in the lottery and the number of rounds in the lottery. You may assume that $1 \le n \le 10000$ and $1 \le m \le 30$.

The following n lines contain the description of the tickets bought by the participant. The i^{th} such line contains m non-negative integers $c_1, ..., c_m$, where c_j ($1 \le j \le m$) is the amount of tickets of round j bought by partipant i. The total number of tickets sold in each round is between 1 and 10^9.

The input ends with a line containing 2 zeros.

Output Specification

For each test case, print n lines of output, where line i contains the probability as a reduced fraction that participant i wins the most money. See the sample output for details.

Sample Input

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

Sample Output

```
1 / 4
1 / 3
5 / 12
0 / 1
0 / 1
1 / 1
```

11629 - Ballot evaluation

Time limit: 1.000 seconds

Before the 2009 elections at the European Parliament, Bill and Ted have asked their friends to make guesses about the outcome of the ballot. Now, the results have been published, so Bill and Ted want to check who was right. But checking the results of their many friends would take a very long time, and they need the evaluation to be done by a computer. Since they are not so good at programming, they ask you for help.

Input Specification

The data provided by Bill and Ted has the following format: The first line consists of the number p of parties followed by the number g of guesses (with $1 \leq p \leq 50$ and $1 \leq g \leq 10000$). Then follow p lines, each line consisting of a unique party name of length ≤ 20 (only containing letters a-z, A-Z and digits 0-9) and the achieved vote percentage of this party with one digit after the decimal point. After the parties follow g lines, each consisting of a guess. A guess has the form $P_1 + P_2 + ... + P_k$ *COMP* n, where P_1 to P_k are party names, *COMP* is one of the comparison operators <, >, <=, >= or = and n is an integer between 0 and 100, inclusively. Each party name occurs at most once in each guess.

Output Specification

For each guess, sum up the vote percentages of the parties and compare them with the specified integer n. Then, print a line stating whether the guess was correct. See the sample output for details.

Sample Input

```
6 5
CDU 30.7
SPD 20.8
Gruene 12.1
FDP 11.0
DIELINKE 7.5
CSU 7.2
FDP > 11
CDU + SPD < 50
SPD + CSU >= 28
FDP + SPD + CDU <= 42
CDU + FDP + SPD + DIELINKE = 70
```

Sample Output

```
Guess #1 was incorrect.
Guess #2 was incorrect.
Guess #3 was correct.
Guess #4 was incorrect.
Guess #5 was correct.
```

Be careful with the comparison of floating point values, because some values in the input (like 0.1) do not have an exact representation as a floating point number.

11631 - Dark roads

Time limit: 2.000 seconds

Economic times these days are tough, even in Byteland. To reduce the operating costs, the government of Byteland has decided to optimize the road lighting. Till now every road was illuminated all night long, which costs 1 Bytelandian Dollar per meter and day. To save money, they decided to no longer illuminate every road, but to switch off the road lighting of some streets. To make sure that the inhabitants of Byteland still feel safe, they want to optimize the lighting in such a way, that after darkening some streets at night, there will still be at least one illuminated path from every junction in Byteland to every other junction.

What is the maximum daily amount of money the government of Byteland can save, without making their inhabitants feel unsafe?

Input Specification

The input file contains several test cases. Each test case starts with two numbers m and n, the number of junctions in Byteland and the number of roads in Byteland, respectively. Input is terminated by $m=n=0$. Otherwise, $1 \leq m \leq 200000$ and $m-1 \leq n \leq 200000$. Then follow n integer triples x, y, z specifying that there will be a bidirectional road between x and y with length z meters ($0 \leq x, y < m$ and $x \neq y$). The graph specified by each test case is connected. The total length of all roads in each test case is less than 2^{31}.

Output Specification

For each test case print one line containing the maximum daily amount the government can save.

Sample Input

```
7 11
0 1 7
0 3 5
1 2 8
1 3 9
1 4 7
2 4 5
3 4 15
3 5 6
4 5 8
4 6 9
5 6 11
0 0
```

Sample Output

```
51
```

11634 - Generate random numbers

Time limit: 1.000 seconds

John von Neumann suggested in 1946 a method to create a sequence of pseudo-random numbers. His idea is known as the "middle-square"-method and works as follows: We choose an initial value a_0, which has a decimal representation of length at most n. We then multiply the value a_0 by itself, add leading zeros until we get a decimal representation of length $2 \times n$ and take the middle n digits to form a_i. This process is repeated for each a_i with $i>0$. In this problem we use $n = 4$.

Example 1: a_0=5555, a_0^2=30858025, a_1=8580,...

Example 2: a_0=1111, a_0^2=01234321, a_1=2343,...

Unfortunately, this random number generator is not very good. When started with an initial value it does not produce all other numbers with the same number of digits.

Your task is to check for a given initial value a_0 how many different numbers are produced.

Input Specification

The input contains several test cases. Each test case consists of one line containing a_0 ($0 < a_0 < 10000$). Numbers are possibly padded with leading zeros such that each number consists of exactly 4 digits. The input is terminated with a line containing the value 0.

Output Specification

For each test case, print a line containing the number of different values a_i produced by this random number generator when started with the given value a_0. Note that a_0 should also be counted.

Sample Input

```
5555
0815
6239
0
```

Sample Output

```
32
17
111
```

Note that the third test case has the maximum number of different values among all possible inputs.

11635 - Hotel booking

Time limit: 4.000 seconds

A transport company often needs to deliver goods from one city to another city. The transport company has made a special deal with a hotel chain which allows its drivers to stay in the hotels of this chain for free. Drivers are only allowed to drive up to 10 hours a day. The transport company wants to find a route from the starting city to the destination city such that a driver can always spend the night in one of the hotels of the hotel chain, and that he needs to drive at most 10 hours from one hotel to the next hotel (or the destination). Of course, the number of days needed to deliver the goods should also be minimized.

Input Specification

The input file contains several test cases. Each test case starts with a line containing an integer n, ($2 \le n \le 10000$), the number of cities to be considered when planning the route. For simplicity, cities are numbered from 1 to n, where 1 is the starting city, and n is the destination city. The next line contains an integer h followed by the numbers $c_1, c_2, ..., c_h$ indicating the numbers of the cities where hotels of the hotel chain are located. You may assume that $0 \le h \le min(n, 100)$. The third line of each test case contains an integer m ($1 \le m \le 10^5$), the number of roads to be considered for planning the route. The following m lines describe the roads. Each road is described by a line containing 3 integers a, b, t ($1 \le a, b \le n$ and $1 \le t \le 600$), where a, b are the two cities connected by the road, and t is the time in minutes needed by the driver to drive from one end of the road to the other. Input is terminated by $n = 0$.

Output Specification

For each test case, print one line containing the minimum number of hotels the transport company has to book for a delivery from city 1 to city n. If it is impossible to find a route such that the driver has to drive at most 10 hours per day, print -1 instead.

Sample Input

```
6
3 2 5 3
8
1 2 400
3 2 80
3 4 301
4 5 290
5 6 139
1 3 375
2 5 462
4 6 300
3
0
2
1 2 371
2 3 230
0
```

Sample Output

```
2
-1
```