台大 ACM 2009 培訓班 網路預賽

注意事項

- 本次預賽共有八題，共十二頁
- 比賽時間為五個小時，以培訓網站時間為準
- 比賽方式為個人賽，計分方式與 ACM-ICPC 相同
- 可參考任何書面資料
- 不可參考網路資源
- 舞弊者永久取消代表台大參加 ACM-ICPC 競賽之資格
- 最快於七月二十日公布培訓班錄取名單

Problem A. Minimizing Maximizer

The company Chris Ltd. is preparing a new sorting hardware called Maximizer. Maximizer has *n* inputs numbered from 1 to *n*. Each input represents one integer. Maximizer has one output which represents the maximum value present on Maximizer's inputs.

Maximizer is implemented as a pipeline of sorters *Sorter*(i_1, j_1), ..., *Sorter*(i_k, j_k). Each sorter has *n* inputs and *n* outputs. *Sorter*(*i*, *j*) sorts values on inputs *i*, *i* + 1,..., *j* in non-decreasing order and lets the other inputs pass through unchanged. The *n*-th output of the last sorter is the output of the Maximizer.

An intern (a former ACM contestant) observed that some sorters could be excluded from the pipeline and Maximizer would still produce the correct result. What is the length of the shortest subsequence of the given sequence of sorters in the pipeline still producing correct results for all possible combinations of input values?

Task

Write a program that:

- reads the description of a Maximizer, i.e. the initial sequence of sorters in the pipeline,
- computes the length of the shortest subsequence of the initial sequence of sorters still producing correct results for all possible input data,
- writes the result.

Input

The first line contains a single integer T indicating the number of data sets.
Input consists of several test case, each separated by a blank line. The first line of the file indicates the number of test cases, and it's followed by a blank line.

The first line of each test case contains two integers *n* and *m* ($2 \leq n \leq 50000$, $1 \leq m \leq 500000$) separated by a single space. Integer *n* is the number of inputs and integer *m* is the number of sorters in the pipeline. The initial sequence of sorters is described in the next *m* lines. The *k*-th of these lines contains the parameters of the *k*-th sorter: two integers i_k and j_k ($1 \leq i_k < j_k \leq n$) separated by a single space.

Output

The output consists of only one line containing an integer equal to the length of the shortest subsequence of the initial sequence of sorters still producing correct results for all possible data. Print a blank line between test cases.

Sample Input

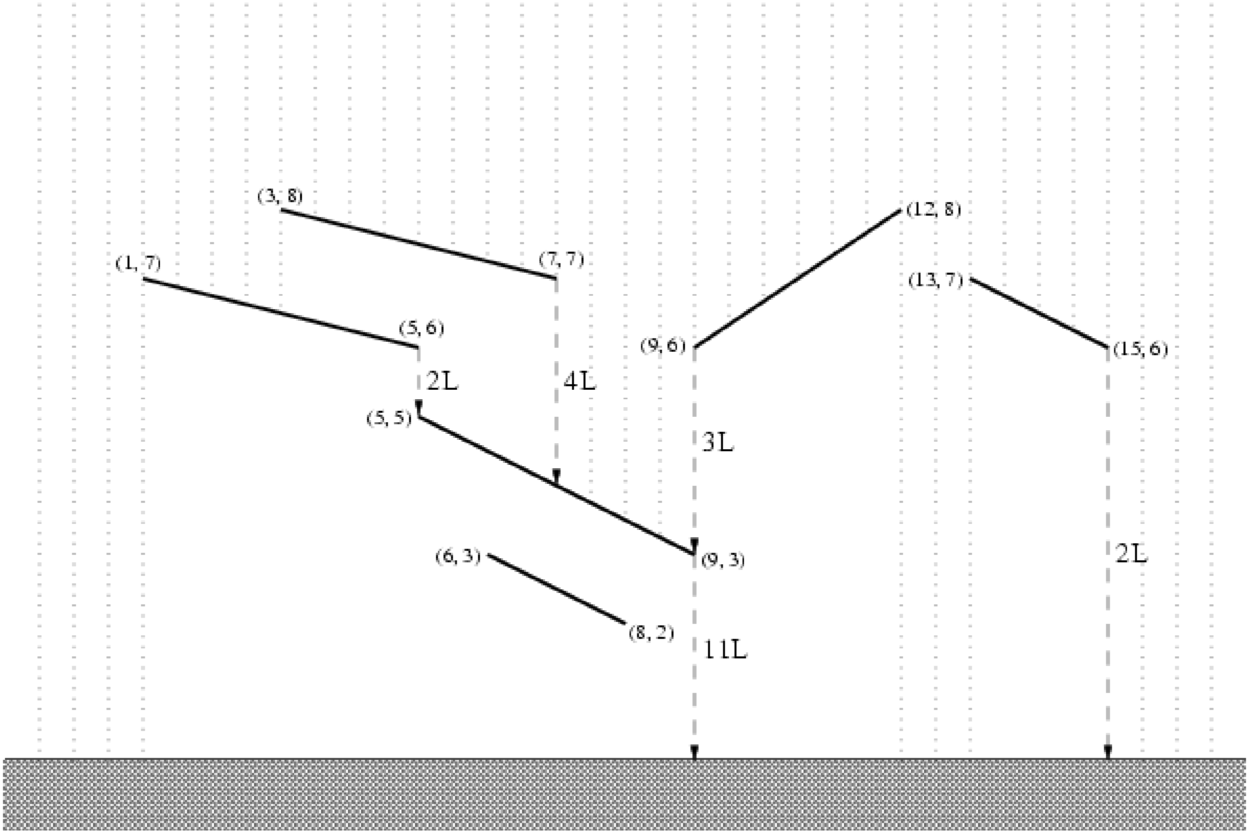
```
1

40 6
20 30
1 10
10 20
20 30
15 25
30 40
```

Sample Output

```
4
```

Problem B. November rain

Contemporary buildings can have very complicated roofs. If we take a vertical section of such a roof it amounts to a number of sloping segments. When it is raining the drops are falling down onto the roof straight from the sky above. Some segments are completely exposed to the rain but there may be some segments partially or even completely shielded by other segments. All the water falling onto a segment flows as a stream straight down from the lower end of the segment on the ground or possibly onto some other segment. In particular if a stream of water is falling on an end of a segment then we consider it to be collected by this segment.



For the purpose of designing a piping system it is desired to compute how much water is flowing down from each segment of the roof. To be prepared for a heavy November rain you should count one liter of rain water falling on a meter of the horizontal plane during one second.

Task

Write a program that:

- reads the description of the roof from the standard input,
- computes the amount of water flowing down in one second from each segment of the roof,
- writes the result to the standard output.

Input

Input consists of several test cases separated by a blank line. The first line of the input indicates the number of test cases, and it's followed by a blank line.

The first line of each dataset contains one integer n ($1 \le n \le 40000$) being the number of segments of the roof. Each of the next n lines describes one segment of the roof and contains four integers x_1, y_1, x_2, y_2 ($0 \le x_1, y_1, x_2, y_2 \le 1000000$, $x_1 < x_2$, y_1 != y_2) separated by single spaces. Integers x_1, y_1 are respectively the horizontal position and the height of the left end of the segment. Integers x_2, y_2 are respectively the horizontal position and the height of the right end of the segment. The segments don't have common points and there are no horizontal segments. Also you can assume that there are at most 100 segments placed above any point on the ground level.

Output

For each test case, the output consists of n lines. The i-th line should contain the amount of water (in liters) flowing down from the i-th segment of the roof during one second.
The outputs of two consecutive cases will be separated by a blank line.

Sample Input

```
6
13 7 15 6
3 8 7 7
1 7 5 6
5 5 9 3
6 3 8 2
9 6 12 8
```

Sample Output

```
2
4
2
11
0
3
```

Problem C. Anagram Division

Given a string **s** and a positive integer **d** you have to determine how many permutations of **s** are divisible by **d**.

Input

First line of the input contains one integer **T** the number of test cases. Each of the test cases is one line containing **s** and **d** separated by a single space.

Output

For each test case output contains an integer the number of permutations of **s** that are divisible by **d**.

Sample Input

```
3
000 1
1234567890 1
123434 2
```

Sample Output

```
1
3628800
90
```

Problem D. Play on Words

Some of the secret doors contain a very interesting word puzzle. The team of archaeologists has to solve it to open that doors. Because there is no other way to open the doors, the puzzle is very important for us.

There is a large number of magnetic plates on every door. Every plate has one word written on it. The plates must be arranged into a sequence in such a way that every word begins with the same letter as the previous word ends. For example, the word ``acm" can be followed by the word ``motorola". Your task is to write a computer program that will read the list of words and determine whether it is possible to arrange all of the plates in a sequence (according to the given rule) and consequently to open the door.

Input

The input consists of T test cases. The number of them (T) is given on the first line of the input file. Each test case begins with a line containing a single integer number N that indicates the number of plates (*1 <= N <= 100000*). Then exactly N lines follow, each containing a single word. Each word contains at least two and at most 1000 lowercase characters, that means only letters 'a' through 'z' will appear in the word. The same word may appear several times in the list.

Output

Your program has to determine whether it is possible to arrange all the plates in a sequence such that the first letter of each word is equal to the last letter of the previous word. All the plates from the list must be used, each exactly once. The words mentioned several times must be used that number of times.
If there exists such an ordering of plates, your program should print the sentence "`Ordering is possible.`". Otherwise, output the sentence "`The door cannot be opened.`".

Sample Input

```
3
2
acm
ibm
3
acm
malform
mouse
2
ok
ok
```

Sample Output

```
The door cannot be opened.
Ordering is possible.
The door cannot be opened.
```

Problem E. Bitwise Kingdom

In the Bitwise Kingdom, located somewhere in the universe, there are exactly 2^N citizens living and each of them has a unique identification string that represents his or her class in the society. An identification string is a binary string of length *N* which consists of characters ‘0’ or ‘1’. The order of classes is defined among the citizens by the following criteria:

1. Citizens identified by a string containing a greater number of ones are ranked higher. For example, “011” indicates a higher class than “100”.
2. 2. Among those who have identification strings with the same number of ones, citizens identified by a lexicographically greater identification string are ranked higher. For example, “110” indicates a higher class than “101”.

For example, if N = 3, there are 8 (= 2^3) people in the country, and their identification strings are “000”, “001”, “010”, “100”, “011”, “101”, “110”, and “111” (from the lowest class to the highest).
You are given two numbers *N* ($1 \leqq N \leqq 60$) and *M* ($1 \leqq M \leqq 2^N$), and you want to resolve the identification string of the person of the *M*-th lowest class among 2^N citizens. Can you write a program to solve this problem?

Input

The input consists of multiple datasets.

Each dataset consists of a line which contains two integers *N* and *M* in this order, separated with a single space. The input does not contain any other extra characters such as leading or trailing spaces.

The end of input is indicated by a line with two zeros. This line is not part of any datasets.

Output

For each dataset, print the identification string of the person of the *M*-th lowest class in one line. Your program may not omit any leading zeros in the answer.

Sample Input

```
3 3
3 5
0 0
```

Sample Output

```
010
011
```

Problem F. Reverse a Road

Andrew R. Klein resides in the city of Yanwoe, and goes to his working place in this city every weekday. He has been totally annoyed with the road traffic of this city. All the roads in this city are one-way, so he has to drive a longer way than he thinks he need.

One day, the following thought has come up to Andrew's mind: "How about making the sign of one road indicate the opposite direction? I think my act won't be out as long as I change just one sign. Well, of course I want to make my route to the working place shorter as much as possible. Which road should I alter the direction of?" What a clever guy he is.

You are asked by Andrew to write a program that finds the shortest route when the direction of up to one road is allowed to be altered. You don't have to worry about the penalty for complicity, because you resides in a different country from Andrew and cannot be punished by the law of his country. So just help him!

Input

The input consists of a series of datasets, each of which is formatted as follows:

N
S T
M
A_1 B_1
A_2 B_2
$: : :$
A_M B_M

N denotes the number of points. S and T indicate the points where Andrew's home and working place are located respectively. M denotes the number of roads. Finally, A_i and B_i indicate the starting and ending points of the i-th road respectively. Each point is identified by a unique number from 1 to N. Some roads may start and end at the same point. Also, there may be more than one road connecting the same pair of starting and ending points.

You may assume all the following: $1 \leqq N \leqq 1000$, $1 \leqq M \leqq 10000$, and $S \neq T$. The input is terminated by a line that contains a single zero. This is not part of any dataset, and hence should not be processed.

Output

For each dataset, print a line that contains the shortest distance (counted by the number of passed roads) and the road number whose direction should be altered. If there are multiple ways to obtain the shortest distance, choose one with the smallest road number. If no direction change results in a shorter route, print 0 as the road number.

Separate the distance and the road number by a single space. No extra characters are allowed.

Sample Input

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

Sample Output

```
1 4
```

Problem G. The problem is as hard as that teaching assistant(s) can't solve

The problem is as hard as that teaching assistant(s) can't solve.

The problem description is quite simple, but unfortunately the solution is unpredictable hard.

Let f(x) = floor(3((ln x)+3)^3), for a given N, your task is to compute the minimum integer x such that f(x) = N.

Input

The first line of the input contains an integer T, where T is the number of the testcase.
There are following T lines, each line of the input contains one integer N, where 0 <= N <= 40000.

Output

For each testcase, output a minimum integer x such that f(x) = N.
If there is no such x, output "BEE" instead of the error.

Sample Input

```
2
10
81
```

Sample Output

```
BEE
1
```

Hint

If you questioned the hardness of the problem, please contacts the (ex-)Teaching Assistant.

Problem H. The problem is as easy as that EVERYONE can solve

The problem is as easy as that someone can solve in 1 minute.

Although the problem is quite simple, unfortunately the problem description is unpredictable hard.

MapReduce is a distributed framework introduced by Google since 2004, for computing certain kinds of distributable problems using a large number of computing nodes. MapReduce classified its computing nodes (WORKERs) into MAPPERs and REDUCERs. Due to the network structure of MapReduce is still too complex, some kind of engineer are going to design a simple protocol between Mappers and Reducers, called Ximple MapReduce.

In Ximple MapReduce network, the original data are stored in some Reducers. In each computing phase, each Reducer reduce the data he stored and received from Mappers, then reduces the data and passes the result to another Mapper. Each Mapper received the data from one Reducer, and the only thing he do is passing the data to another Reducer. In each phase, every worker can only do one job. Finally, all data need to be reduced and passed to Central Mapper with ID 0.

Unfortunately, Ximple MapReduce is not much reliable between its workers. If some worker crashes, the total computing got failures. Engineer now can detect the failure, but he need to know the structure network of this computing phase to rebuild computing. One of the method is to take a SNAPSHOT of this network. The snapshot is some characteristic of the network, and a reasonable approach is to compute the determinant of the network, which is defined by the determinant of its adjacency matrix.

Hence, till now, our Engineer are still working on computing the determinant of the network. Because this is not a easy procedure, could you write a program to help our Engineer?

Input

The first line of the input file contains an integer T indicating the number of test cases to follow.
T <= 20.

Each test case contains exactly one line. The first input of each test cases is an integer N, indicating number of workers in the network, N <= 300. All worker with ID between 0 and N-1. Mappers has even ID and Reducers has odd one.

There are following N-1 integers P1, P2, ... Pi, ..., P_(N-1). The worker i has to pass its data to the worker Pi. The worker with ID 0 is the Central Mapper, it has no need no pass the result to other workers.

Output

For each test case, output the determinant of the network in the test case.

Sample Input

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

Sample Output

```
-1
-1
```

Hint

1. The adjacency matrix A of a network is defined as [a_i,j] with 0 <= i, j <= N-1.
The worker i pass the data to worker j iff a_i,j = a_j,i = 1.

2. For the determinant of a matrix, please view HERE

3. Each test case must be a legal Ximple Mapredce Network.