Kharagpur Site Online Problems 2013

Problem #1: List Editing

At the ACME University, as part of the Graduate course work, each student is required to undergo project work. The authorities always make an effort to find a match among the students' interests and the interests of the faculty members. First, they take a list of topics from the students' and a list of topics from the faculty. In order to avoid identifying students and faculties, all they preserve is simply, two consolidated lists of topics – one from the students and one from the faculties. No personal identification exists within these lists. The lists are sorted in ascending order of the indices of project topics in the repository. No topic appears twice in the same list.

To make the process simple, ACME University maintains a repository of 100 project topics - numbered from 1 to 100, inclusive. The students and faculty members must choose among those topics only. Their lists may have different number of topics; since more than one student (or faculty) may choose to work work on the same topic or some may like to work on multiple topics etc.

Once these two lists have been obtained, the university tries to make them identical. The student's list is modified to match with the faculty's list. The modification may be via

- inserting a new topic in the list
- deleting a topic from the list
- substituting a topic by another one

The university wants to do this with minimal number of modifications to the list as each modification needs one full day to be communicated and accepted by the students. It is assumed that the students do not disagree to any such modification.

Write a program that accepts the lists from students and faculties and computes the minimum number of modifications that will require to be made to make the student's list identical to the faculty's list.

Input

The input contains several test cases. The first line of the input contains the number T, the number of test cases. The first line in each test case contains the number M, the number of topics in the students' list. The second line contains M integers A_1, A_2 .., A_M separated by single space characters. The third line in each test case contains the number N, the number of topics in the faculty's list. The fourth line contains N integers B_1, B_2 .., B_N separated by single space characters.

Output

Output T numbers, each on a line by itself, without any leading spaces or zeros. The number should be the minimum number of modifications that can be made to the students' list to make it identical to the faculty's list, for each respective test case.

Constraints

$1 \leq T \leq 100$
$1 \leq M, N \leq 100$
$1 \leq A_i, B_i \leq 100$
The list $\{ A_1 .. A_M \}$ has no duplicates.
The list $\{ A_1 .. A_M \}$ is sorted in ascending order.
The list $\{ B_1 .. B_N \}$ has no duplicates.
The list $\{ B_1 .. B_N \}$ is sorted in ascending order.

Sample Input

```
2
4
1 3 5 7
2
4 6
3
1 2 3
2
2 3
```

Sample Output

```
4
1
```

Problem #2: Jackpot

A certain business farm has introduced a loyalty program.

At the time of registration to the loyalty program, a customer is given 1 point. The points will get upgraded after the first new purchase every week. After such purchase, the customer can select between the following two policies to upgrade points:

- Get exactly 1 more point
- Double the points that he/she has

For example, if the customer had 43 points before a purchase, he/she can upgrade it to 44 or 86, depending upon points upgrade policy of their choosing.

A certain business farm has introduced a loyalty program.

At the time of registration to the loyalty program, a customer is given 1 point. The points will get upgraded after the first new purchase every week. After such purchase, the customer can select between the following two policies to upgrade points:

- Get exactly 1 more point
- Double the points that he/she has

For example, if the customer had 43 points before a purchase, he/she can upgrade it to 44 or 86, depending upon points upgrade policy of their choosing.

At the time of registration, each customer is also given a **Jackpot** target. Special facilities will be provided to the customer if the accumulated points of the customer become equal to this Jackpot. However, accumulating points more or less than the Jackpot does not earn them anything.

For example, for a customer with Jackpot target 437, special facilities will be provided if and only if his/her points becomes exactly equal to 437 after some purchases.

Of course the simplest plan that may be adopted by the customer is to increment his points by 1 after each purchase. However, an intelligent customer would like to achieve the Jackpot score in the minimum number of purchases.

Even still, for a single customer, the number of weeks in which he gets the Jackpot will be equal to the number of purchases, since only the first purchase in a week qualifies towards upgrade of points.

The most intelligent customer of all would partner with a friend and try to merge their points in the end. Of course this is better because this means that fewer weeks will be necessary to achieve the Jackpot. Alice and Bob are going to use this strategy. They will decide a sub-target score for

themselves (which may be different for each of them) and try to reach that sub-target in the fewest number of weeks (at the rate of one purchase per week). Once both of them reach that sub-target, they will merge their points to achieve Alice's Jackpot target! See the Explanation Section for clarification.

Write a program that takes as input Alice's Jackpot score and outputs the minimum number of weeks Alice and Bob take to achieve the Jackpot score. Note that Alice and Bob will never merge their scores in between. They will only merge their scores once they reach the respective sub-targets they decide for themselves.

Input

The first line of input contains the number T, the number of test case. Each test case contains a single integer N, Alice's Jackpot target.

Output

For each test case, output a single line containing the number W, the minimum number of weeks necessary for Alice and Bob to achieve Alice's Jackpot target.

Constraints

$1 \leq T \leq 1000$
$2 \leq N \leq 1000000000$

Sample Input

```
4
2
4
10
55
```

Sample Output

```
0
1
3
6
```

Problem #3: The K-th Path

WARNING: The Input files may be as large as 8 MB. Use faster I/O methods.

There is a tree, consisting of N nodes. A path is a sequence of one or more nodes of this tree, where every two adjacent nodes are connected by an edge and no node is visited twice. Generally, there are N^2 paths. Note that we are counting paths from each node onto itself as well. One path is smaller than another if the sequence of nodes of the first path is lexicographically smaller than the sequence of nodes of the second path (by usual rules).

Thus (1,3,2) will be smaller than (1,3,4,5). And (2,4) will be smaller than (3).

Every day, Lucy writes out a path on a sheet of paper. On the first day, she writes the lexicographically smallest path. On the second day, she writes the lexicographically second smallest path, and so on. Generally, on the K^{th} day, Lucy writes out the lexicographically K^{th} smallest path.

Nana likes to read Lucy's notes about the paths in the tree. She is curious about the path that will be written on the Q^{th} day. Could you please help her?

Input

The first line of input consists of a single integer T, the number of the test cases. Then, there are T test cases, given in the following form: the first line of the test case consists of an integer N, the number of nodes and an integer Q, the number from the statement. Then, N-1 lines follow. Each such line will consist of two integers X and Y with the meaning that there's an edge, connecting nodes X and Y in the tree. You may assume that you are always given a valid tree.

Output

For every test case, output the lexicographically Q^{th} smallest path in the given tree on a single line, without leading or trailing spaces. There should be exactly one space between successive nodes in the path. Output the answer to the t^{th} test case should be given on the t^{th} line.

Constraints

$1 \leq N \leq 100000$
$1 \leq Q \leq N^2$
1 ≤ Sum of N over all the test cases in the single file ≤ 1000000

Sample Input

```
3
6 21
1 2
2 3
2 4
2 5
2 6
7 35
1 2
1 3
3 4
4 5
4 6
4 7
6 16
1 2
1 3
1 4
2 5
1 6
```

Sample Output

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

Problem #4: Substring Problem

How many times does the integer K occur in the sequence obtained by writing out all the natural numbers from 1 to N, without any whitespace in between.

For example, if N = 20, the sequence would look like

```
12345678910111213141516171819 20
```

In the above sequence, the integer 12 occurs exactly 2 times. Also, the integer 11 appears exactly 2 times. Note that the two occurances of 11 overlap, but should still be counted separately.

How many times does the integer K occur in the sequence obtained by writing out all the natural numbers from 1 to N, without any whitespace in between.

For example, if N = 20, the sequence would look like

```
12345678910111213141516171819 20
```

In the above sequence, the integer 12 occurs exactly 2 times. Also, the integer 11 appears exactly 2 times. Note that the two occurances of 11 overlap, but should still be counted separately.

Input

The first line of input consists of an integer T - the number of the test cases. Then, T lines, consisting of two integers N and K follow. N and K are separated by a single space.

Output

For every test case, output the number of times that the integer K appears in the string, obtained by writing out the first N natural numbers.

Constraints

```
1 ≤ N, K ≤ 1000000
1 ≤ Sum of N over all the test cases ≤ 1000000
```

Sample Input

```
5
100 333
97 5
4 5
11 1
10 1
```

Sample Output

```
1
20
0
4
2
```