E. Thoroughly Bureaucratic Organization

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Once n people simultaneously signed in to the reception at the recently opened, but already thoroughly bureaucratic organization (abbreviated TBO). As the organization is thoroughly bureaucratic, it can accept and cater for exactly one person per day. As a consequence, each of n people made an appointment on one of the next n days, and no two persons have an appointment on the same day.

However, the organization workers are very irresponsible about their job, so none of the signed in people was told the exact date of the appointment. The only way to know when people should come is to write some requests to TBO.

The request form consists of m empty lines. Into each of these lines the name of a signed in person can be written (it can be left blank as well). Writing a person's name in the same form twice is forbidden, such requests are ignored. TBO responds very quickly to written requests, but the reply format is of very poor quality — that is, the response contains the correct appointment dates for all people from the request form, but the dates are in completely random order. Responds to all requests arrive simultaneously at the end of the day (each response specifies the request that it answers).

Fortunately, you aren't among these n lucky guys. As an observer, you have the following task — given n and m, determine the minimum number of requests to submit to TBO to clearly determine the appointment date for each person.

Input

The first line contains a single integer t $(1 \leq t \leq 1000)$ — the number of test cases. Each of the following t lines contains two integers n and m $(1 \leq n, m \leq 10^9)$ — the number of people who have got an appointment at TBO and the number of empty lines in the request form, correspondingly.

Output

Print t lines, each containing an answer for the corresponding test case (in the order they are given in the input) — the minimum number of requests to submit to TBO.

Examples

input

```
5
4 1
4 2
7 3
1 1
42 7
```

output

```
3
2
3
0
11
```

Note

In the first sample, you need to submit three requests to TBO with three different names. When you learn the appointment dates of three people out of four, you can find out the fourth person's date by elimination, so you do not need a fourth request.

In the second sample you need only two requests. Let's number the persons from 1 to 4 and mention persons 1 and 2 in the first request and persons 1 and 3 in the second request. It is easy to see that after that we can clearly determine each person's appointment date regardless of the answers obtained from TBO.

In the fourth sample only one person signed up for an appointment. He doesn't need to submit any requests — his appointment date is tomorrow.