6873 Locked Treasure

Picture from Wikimedia Commons

A group of n ($1 \le n \le 30$) bandits hid their stolen treasure in a room. The treasure needs to be locked away until there is a need to retrieve it. Since the bandits do not trust each other, they wanted to ensure that at least m ($1 \le m \le n$) of the bandits must agree in order to retrieve the treasure.

They have decided to place multiple locks on the door such that the door can be opened if and only if all the locks are opened. Each lock may have up to n keys, distributed to a subset of the bandits. A group of bandits can open a particular lock if and only if someone in the group has a key to that lock.

Given n and m, how many locks are needed such that if the keys to the locks are distributed to the bandits properly, then every group of bandits of size at least m can open all the locks, and no smaller group of bandits can open all the locks?

For example, if $n = 3$ and $m = 2$, only 3 locks are needed — keys to lock 1 can be given to bandits 1 and 2, keys to lock 2 can be given to bandits 1 and 3, and keys to lock 3 can be given to bandits 2 and 3. No single bandit can open all the locks, but any group of 2 bandits can open all the locks. You should also convince yourself that it is not possible to satisfy the requirements with only 2 locks.

Input

The first line of input contains a positive integer indicating the number of cases to follow. Each case is specified by the two integers n and m on one line, separated by a single space.

Output

For each case, print a line containing the number of locks needed, a single space, and the number of keys each bandit will need. No leading or trailing whitespace is to appear on an output line.

Sample Input

```
4
3 2
5 1
10 7
5 3
```

Sample Output

```
3 2
1 1
210 84
10 6
```