

Toss is an important part of any event. When everything becomes equal toss is the ultimate decider. Normally a fair coin is used for Toss. A coin has two sides head(H) and tail(T). Superstition may work in case of choosing head or tail. If anyone becomes winner choosing head he always wants to choose head. Nobody believes that his winning chance is 50-50. However in this problem we will deal with a fair coin and n times tossing of such a coin. The result of such a tossing can be represented by a string. Such as if 3 times tossing is used then there are possible 8 outcomes.

HHH HHT HTH HTT THH THT TTH TTT

As the coin is fair we can consider that the probability of each outcome is also equal. For simplicity we can consider that if the same thing is repeated 8 times we can expect to get each possible sequence once.

In the above example we see 1 sequence has 3 consecutive H, 3 sequence has 2 consecutive H and 7 sequence has at least single H. You have to generalize it. Suppose a coin is tossed n times. And the same process is repeated 2^n times. How many sequence you will get which contains a sequence of H of length at least k .

Input

The input will start with two positive integer, n and k ($1 \leq k \leq n \leq 100$). Input is terminated by EOF.

Output

For each test case show the result in a line as specified in the problem statement.

Sample Input

```
4 1
4 2
4 3
4 4
6 2
```

Sample Output

```
15
8
3
1
43
```