165 Stamps — Modified from UVA165

The government of Nova Mareterrania requires that various legal documents have stamps attached to them so that the government can derive revenue from them. In terms of recent legislation, each class of document is limited in the number of stamps that may be attached to it. The government wishes to know how many different stamps, and of what values, they need to print to allow the widest choice of values to be made up under these conditions. Stamps are always valued in units of $1.

This has been analysed by government mathematicians who have derived a formula for $n(h, k)$, where h is the number of stamps that may be attached to a document, k is the number of denominations of stamps available, and n is the largest attainable value in a continuous sequence starting from $1. For instance, if $h = 3$, $k = 2$ and the denominations are $1 and $4, we can make all the values from $1 to $6 (as well as $8, $9 and $12). However with the same values of h and k, but using $1 and $3 stamps we can make all the values from $1 to $7 (as well as $9). This is maximal, so $n(3, 2) = 7$.

Unfortunately the formula relating $n(h, k)$ to h, k and the values of the stamps has been lost — it was published in one of the government reports but no-one can remember which one, and of the three researchers who started to search for the formula, two died of boredom and the third took a job as a lighthouse keeper because it provided more social stimulation.

The task has now been passed on to you. You doubt the existence of a formula in the first place so you decide to write a program that, for given values of h and k, will determine an optimum set of stamps and the value of $n(h, k)$.

Input

Input will consist of several lines, each containing a value for h and k. The file will be terminated by two zeroes (0 0). For technical reasons the sum of h and k is limited to 9. (The President lost his little finger in a shooting accident and cannot count past 9).

Output

The value of $n(h, k)$ for each value of h and k is output on one line.

Sample Input

```
3 2
0 0
```

Sample Output

```
7
```