

A: ASP

Source file name: `asp.c`, `asp.cpp`, `asp.java`, or `asp.py`

Author: Rodrigo Cardoso

John is worried about the passwords he uses for his internet activities. He decided to build his passwords according to some rules that he considers safe enough. He calls each one of these passwords an *asp* (for a *safe password*).

Suppose that A is a given asp. A *subasp* of A is any word that corresponds to a sequence of two or more consecutive symbols in A .

John's conditions for an asp A are rather simple:

- A is a word built from an alphabet of N symbols.
- It is not possible to find a subasp of A more than once within it.
- No two consecutive symbols in A are the same.

For instance, if the given alphabet is $\{a, b, c, d\}$, it is possible to build asps like *abac*, *abcdbadb* and *bcbadb*. On the other hand, *baac*, *abcdcabedb* are not asps.

John wants to know how long an asp could be, given the alphabet's length. For instance, if asps were made with only $\{a, b\}$, a longest asp could have at most three symbols, e.g., *aba*. You are asked to help John with his task.

Input

The input consists of several test cases. A case is defined with a line with a positive integer number N , $1 < N < 1000$, the number of symbols in the alphabet used to build asps. The end of the input is signaled with a line containing a single 0, which should be not processed.

The input must be read from standard input.

Output

For each test case, output a line with one integer value, corresponding to the length of an asp of maximal length that may be built with an alphabet of N symbols.

The output must be written to standard output.

Sample Input	Sample Output
2	3
3	7
0	