

## First and oldest application

The idea for this program came from one experiment which was made in the beginning of the twenty century. The idea of this experiment was to teach the hens to count. One biologist put in a row grains but every second grain in the row was treated with a special chemical which makes it not tasty.



So the row was: one good grain, one bad grain, and so on. The hen started to eat the grain but very soon it understood that the second grain is not tasty and start to eat only the good ones. I.e. the hen started to count: "one - two, one - two". After that the task for the hen was made harder. One good grain and two bad ones. Then the hen started to count "One, two, three. One, two, three." The result was that some hens studded to count to four other to five (hens like people have different intelligence).

The idea was to make the next step and after making hens to count (to catch dependencies) to study computers to do the same. The result was this simple program which can catch most dependencies which the human being can catch.



Like example of a row which is easy for our program is the Fibonacci row. This is the row (1, 1, 2, 3, 5, 8, 13, and so on). To give this row to the program you should first to code it in a row of zeros and ones. The coded row can be the following: (0101 0110 1110 1111 10...) i.e. zero like separator and ones which number give the next number from Fibonacci row.

How this program works. As we said in a very simple way. First it ask you to think for infinite row of zeroes and ones, then you can give some beginning of your row (it can be the empty beginning either). After that our programs look for the first Turing machine which can generate this beginning and its prognoses is the next digit which this Turing machine will generate. This mean that when our program catch your dependence then it can guess the next numbers in the row but also it can show you the rule which generates this row (the Turing machine).

Theoretically our program can guess all calculable rows (with some restriction for effectiveness) but in practice the time for this grows exponentially with the growth of the difficulty and for more complicated rows it thinks practically infinitely. You will ask how intelligent is this program. Well, the answer is: like a human being. The rows like the Fibonacci row and like two on power  $N$  can be easily guessed by people and by this program but more complicated rows practically cannot be guessed by any of them.

**You can download and try [this program](#).** It is an old program and has DOS interface. It was written on Turbo C somewhere between 1991 and 1993. It is one useless but anyway amusing program.