Problem 5—A Million Combinations

Professor Plum likes to teach Discrete Mathematics and one of his favorite expressions is "There must be more than a million combinations". Therefore, it is not too surprising that his favorite Combinatorics problem involves tallying the number of ways to achieve a million combinations.

Typically, he provides his students with the following background information.

> "There are exactly ten ways of selecting three of five things. Consider the digits 12345, ten ways are:
>
> 123, 124, 125, 134, 135, 145, 234, 235, 245, and 345
>
> In combinatorics, we use the notation, ${}^{5}C_{3} = 10$.
>
> In general,
>
> , where
>
> Blaise Pascal (17th century mathematician) developed Pascal's triangle to calculate the number of combinations in a "dynamic programming" fashion. Pascal's triangle is written as:

```
                                                       Row #
            1                                            0
          1   1                                          1
        1   2   1                                        2
      1   3   3   1                                      3
    1   4   6   4   1                                    4
  1   5  10  10   5   1                                  5
            .
            :
```

> It is not until n = 23, that a value exceeds one-million: ${}^{23}C_{10} = 1144066$."

Given a specific integer value *max*, you are to determine how many <*n*,*r*> pairs ($1 \leq r \leq n \leq max$) give values of ${}^{n}C_{r}$ greater than one-million?

The pairs do not necessarily need to give distinct ${}^{n}C_{r}$ values. For example, pairs <23,10> and <23,13> (${}^{23}C_{10} = {}^{23}C_{13} = 1144066$) should each be tallied if *max* is 23 or bigger.

INPUT SPECIFICATION– File name "prob5.in"
The input file contains a single line with a positive integer *max*.

OUTPUT SPECIFICATION.
The output file should contain a single line with the number of <n,r> pairs ($1 \leq r \leq n \leq max$) giving values of ${}^{n}C_{r}$ greater than one-million.

SAMPLE INPUT.
25<EOLN>
<EOF>

SAMPLE·OUTPUT.
21<EOLN>
<EOF>