

1 Problem

There are exactly ten ways of selecting three from five, 12345:

123, 124, 125, 134, 135, 145, 234, 235, 245, and 345

In combinatorics, we use the notation, ${}_5C_3 = 10$.

In general,

$${}_nC_r = \frac{n!}{r!(n-r)!} \quad (1)$$

where $r \leq n$, $n! = n \times (n-1) \times \dots \times 3 \times 2 \times 1$, and $0! = 1$.

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

How many, not necessarily distinct, values of ${}_nC_r$, for $1 \leq n \leq 100$, are greater than one-million?

2 Solution

```
import Data.List
import qualified Data.Map as Map
import Data.Maybe
import System.Environment
import qualified Data.Set as Set
import Data.Numbers
import Data.Numbers.Primes

factorial :: Integer -> Integer
factorial 0 = 1
factorial n = n * factorial (n - 1)

(!) :: Int -> Int
(!) 0 = (1 :: Int)
(!) 1 = (1 :: Int)
(!) x = x * (!) (x - 1)

choose :: Integer -> Integer -> Integer
choose n r = (factorial n) `div` ((factorial r) * (factorial (n - r)))

permute :: Integer -> Integer -> Integer
permute n r = (factorial n) `div` (factorial (n - r))

main = do
  let combs = concat $ map (\zz -> map (\lz -> choose zz z) [0..zz]) [1..100]
      solns = filter (>= 1000000) combs
      nsols = length solns
  putStrLn $ "There are " ++ show nsols ++ " values of nCr which\n"
    ++ "evaluate to more than 1,000,000."
```

3 Result

```
runhaskell problem53.lhs
There are 4075 values of nCr which
evaluate to more than 1,000,000.
```