module fsieve

import StdClass; // RWS

import StdInt, StdReal

NrOfPrimes :== 3000

primes :: [Int]

primes = pr where pr = [5 : sieve 7 4 pr]

sieve :: Int !Int [Int] -> [Int]

sieve g i prs

| isPrime prs g (toInt (sqrt (toReal g))) = [g : sieve` g i prs]

| otherwise = sieve (g + i) (6 - i) prs

sieve` :: Int Int [Int] -> [Int]

sieve` g i prs = sieve (g + i) (6 - i) prs

isPrime :: [Int] !Int Int -> Bool

isPrime [f:r] pr bd

| f>bd = True

| pr rem f==0 = False

| otherwise = isPrime r pr bd

select :: [x] Int -> x

select [f:r] 1 = f

select [f:r] n = select r (n - 1)

Start :: Int

Start = select [2, 3 : primes] NrOfPrimes