```
-- ------------------------------------------------------------------------------------
-- I1M CURSO 2013-14.
-- PRUEBA 1 DE EVALUACION ALTERNATIVA (6 NOVIEMBRE 2013)
-- ------------------------------------------------------------------------------------
-- NOMBRE Y APELLIDOS:
-- GRUPO:
-- EMAIL:
-- ------------------------------------------------------------------------------------
import Test.QuickCheck

-- EJERCICIO 1.
-- Decimos que dos numeros son hermanos si tienen el mismo numero de
-- divisores propios. Por ejemplo, 6 y 22 son hermanos porque ambos
-- tienen tres divisores propios.
-- EJERCICIO 1.1.
-- Definir una funcion (hermanos x y) que reconozca si x e y son hermanos.
-- Por ejemplo,
-- hermanos 6 10 == True
-- Ya que 6 tiene tres divisores propios y 10 tambien.
-- hermanos 3 4 == False
-- Ya que 3 tiene un divisor propio y 4 tiene dos.

hermanos = undefined

-- EJERCICIO 1.2.
-- Definir una funcion hermanosHasta tal que  (hermanosHasta n) que
-- devuelva la lista de los pares de numeros hermanos menores o iguales
-- a n.
-- Por ejemplo,
-- hermanosHasta 6 ==
-- [(1,1),(2,2),(2,3),(2,5),(3,2),(3,3),(3,5),(4,4),(5,2),(5,3),(5,5),(6,6)]

hermanosHasta = undefined

-- EJERCICIO 1.3.
-- Define una propiedad prop_hermanos1, que compruebe que si (x,y) es un
-- par de numeros hermanos, entonces (y,x) tambien lo es.

prop_hermanos1 = undefined

-- EJERCICIO 1.4.
-- Define una propiedad prop_hermanos2 que compruebe que todo x es
-- hermano de si mismo.

prop_hermanos2 = undefined

-- EJERCICIO 1.5.
-- Define una funcion primerosHermanos tal que (primerosHermanos k)
-- los primeros k pares de numeros hermanos evitando duplicar, es decir,
-- evitando los pares (x,x) y si aparece el par (x,y) entonces no debe
-- aparecer (y,x).
-- Por ejemplo,
```

-- primerosHermanos 10 ==
-- [(3,2),(5,2),(5,3),(7,2),(7,3),(7,5),(8,6),(9,4),(10,6),(10,8)]

primerosHermanos =undefined

-- ---
-- EJERCICIO 2.
-- Definir una funcion superDiv tal que (superDiv n) construye la lista
-- de listas que contienen los divisores de cada uno de los divisores de
-- n. Por ejemplo,
-- superDiv 10 == [[1],[1,2],[1,5],[1,2,5,10]]
-- superDiv 12 == [[1],[1,2],[1,3],[1,2,4],[1,2,3,6],[1,2,3,4,6,12]]

superDiv = undefined

-- Definir una funcion noPrimos tal que (noPrimos n) devuelve la lista
-- que resulta de sustituir cada elemento de (superDiv n)
-- por el numero de numeros no primos que contiene.
-- Por ejemplo:
-- noPrimos 10 == [0,1,1,2]
-- noPrimos 12 == [0,1,1,1,2,2]

noPrimos = undefined

-- ---
-- EJERCICIO 3.
-- Una lista es genial si la diferencia en valor absoluto entre
-- cualesquiera dos terminos consecutivos es siempre mayor o igual que
-- la posicion del primero de los dos. Por ejemplo,
-- genial [1,3,-4,1] == True
-- Ya que |1-3| = 2 >= 0 = posicion del 1,
-- |3-(-4)|= 7 >= 1 posicion del 3,
-- |(-4)-1|= 5 >= 2 posicion del -4.
-- genial [1,3,0,1,2] == False
-- Ya que 3-1 = 2 >=0; |0-3|=3>=1; pero |1-0|=1 no es >=2.
-- Definir por comprensíon una funcion (genial xs) tal que reconozca si
-- xs es una lista genial.

genial :: [Int] -> Bool
genial = undefined