

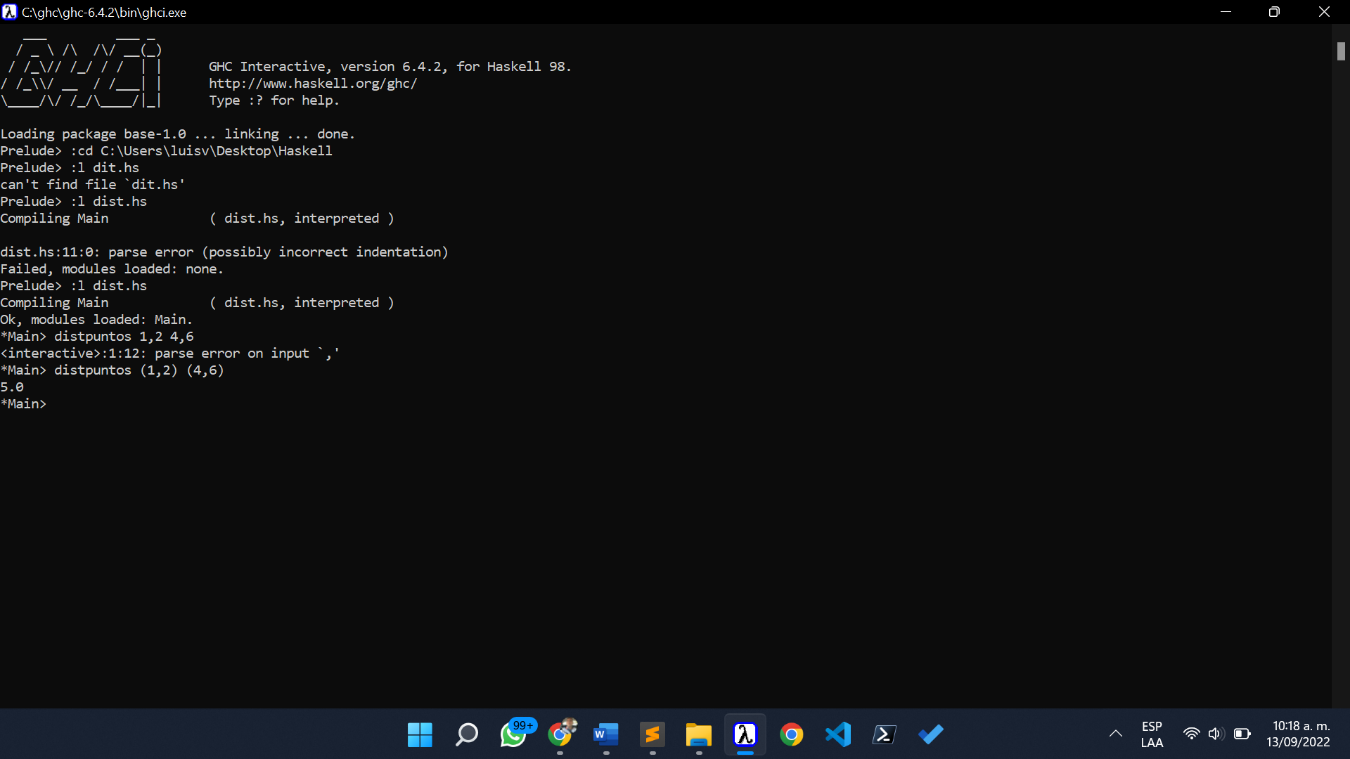
Programación Lógica y Funcional.

Práctica 1.2 Programas en programación funcional Haskell.

Fecha: 13/09/2022.

Alumno: Luis Alberto Vargas González.

1.- Programa que calcule la distancia entre dos puntos utilizando la formula cuadrática.



-- distpuntos :: Integer -> Integer -> Integer -> Integer

x1=1

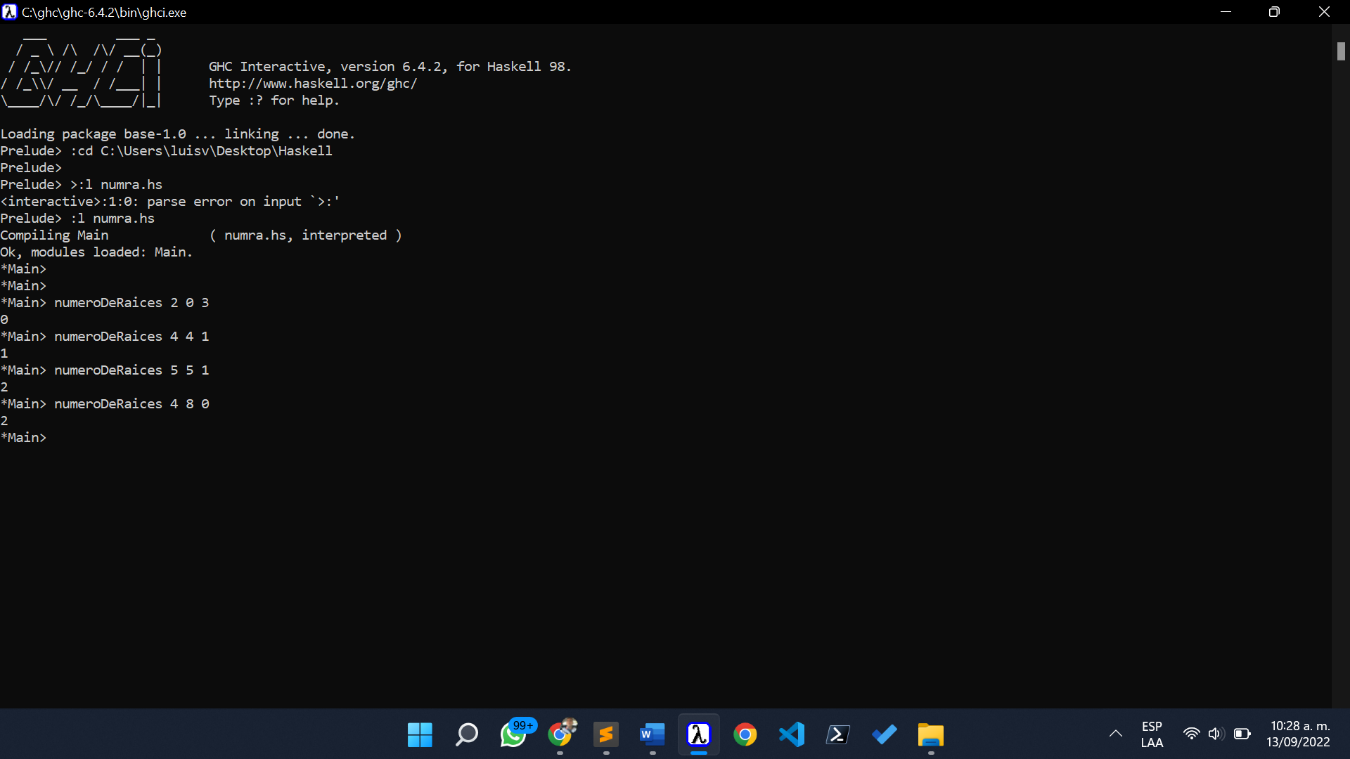
y1=2

x2=4

y2=6

distpuntos (x1,y1) (x2,y2) = sqrt((x1-x2)^2+(y1-y2)^2)

2.- Definir la función raíces de forma que (raíces a b c) devuelve la lista de las raíces reales de la ecuación ax2 + bx + c = 0.



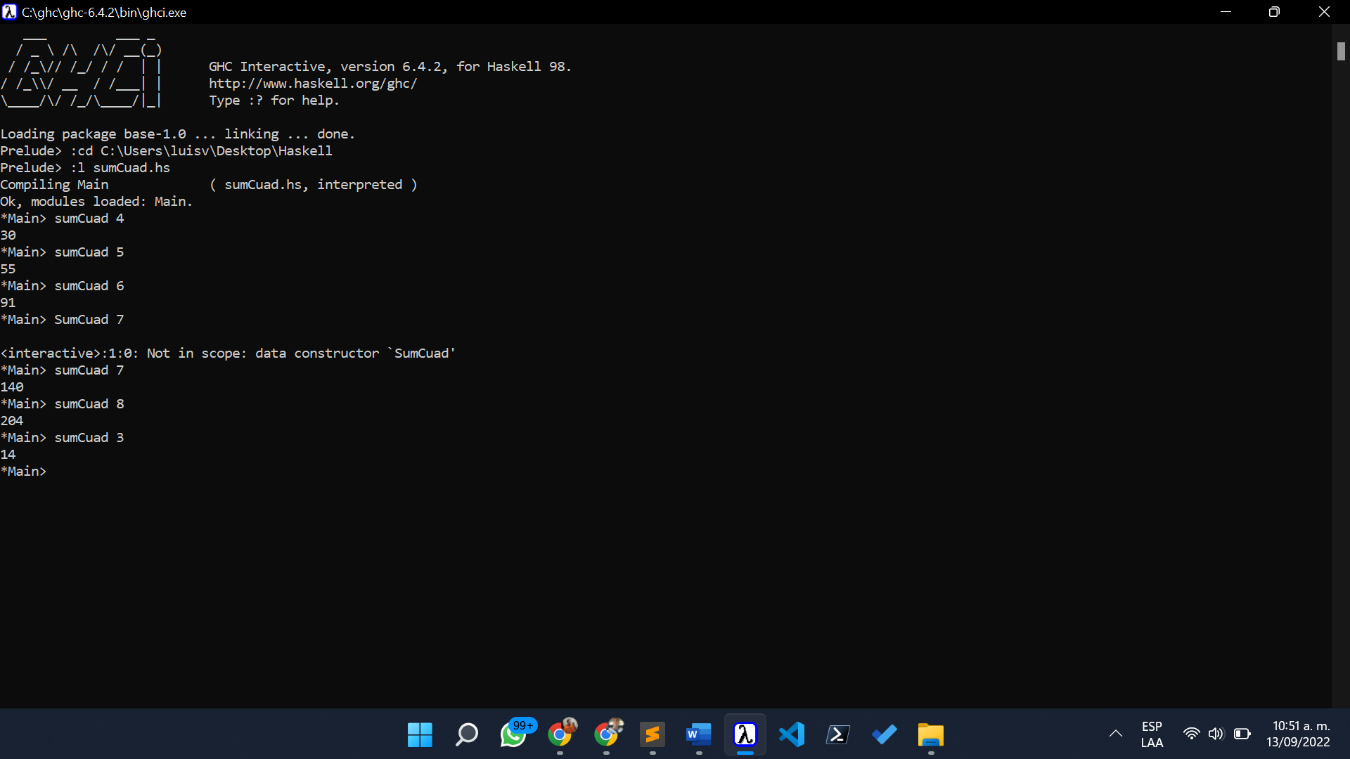
numeroDeRaices a b c | d < 0 = 0

| d == 0 = 1

| otherwise = 2

where d = b^2-4\*a\*c

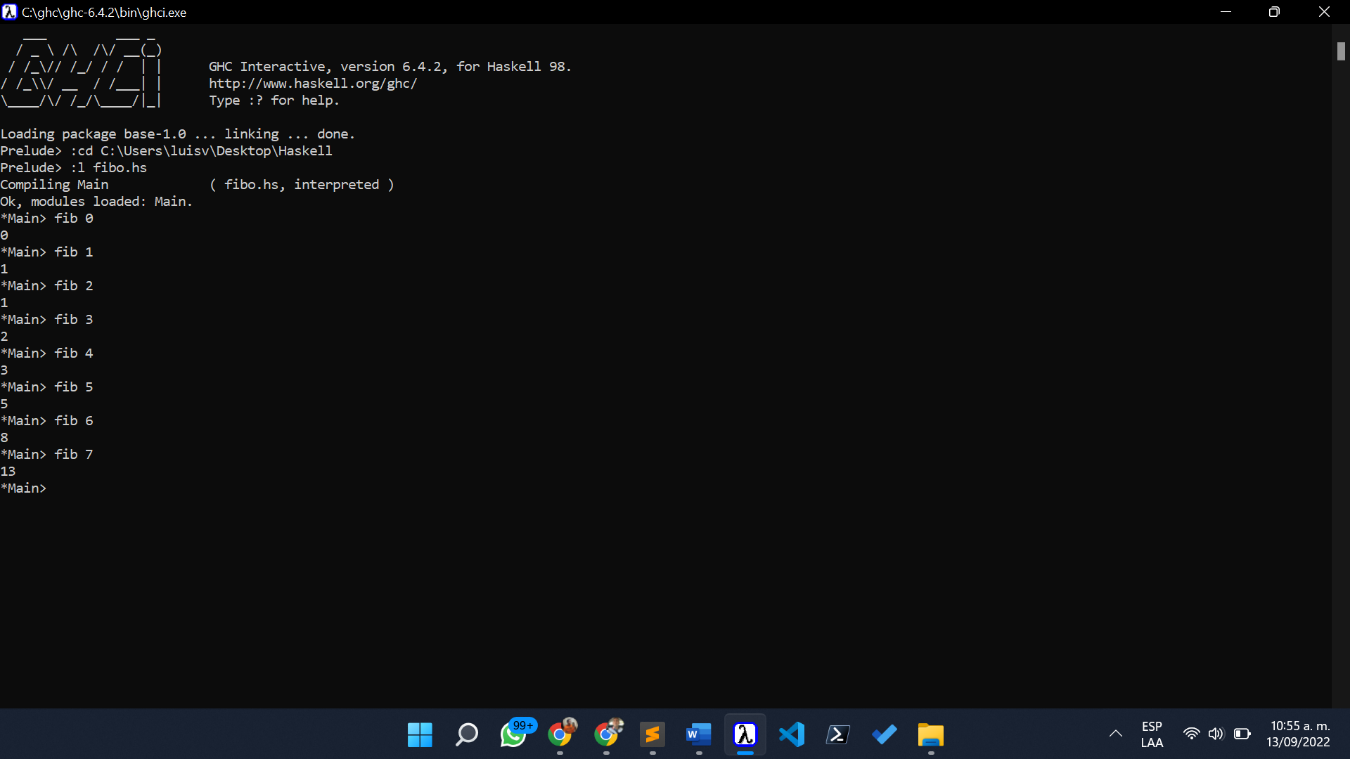
3.- Programa que calcule los cuadrados de los primeros n números.



sumCuad :: Integer -> Integer

sumCuad n = sum [x^2 | x <- [1..n]]

4.- Programa que determine dado un numero n, determine el Fibonacci de n.



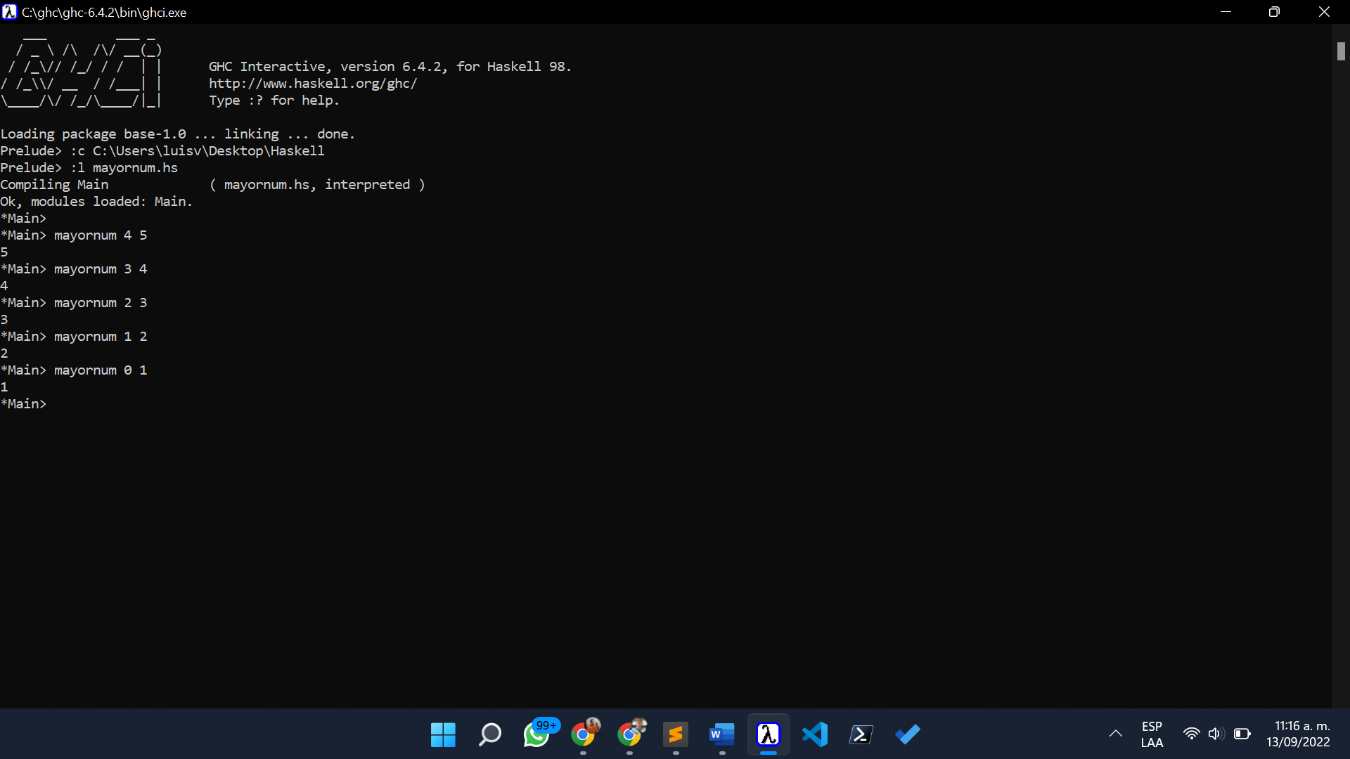
fib :: Integer -> Integer

fib 0 = 0

fib 1 = 1

fib n = fib (n-1) + fib (n-2)

5.- Programa que determine el mayor de dos números.



mayornum :: Integer -> Integer -> Integer

mayornum a b

| a > b = a

| otherwise = b

**Conclusiones.**

Por último, podemos concluir que; en definitiva no se puede programar líneas de código haskell o cualquier lenguaje de tipo declarativo teniendo en mente , la estructura y definiciones de los lenguajes imperativos y de los paradigmas por igual, por lo que el hacer de dicha manera genera muchos errores y confusiones al programador , asi como errores de compilación y de generación del código del lenguaje.

Adicionalmente pudimos observar como es que las declaraciones de este lenguaje de programación son cientos de veces más sencillas y resumidas que las hechas en lenguajes de tipo imperativo, logrando asi una mayor comprensión y desarrollo mas dinámico de la lógica del programador.