Practica 2

Trabajando con enteros

productoPunto3D

Crea una funcion llamada **productoPunto3D** que recibe dos tripletas que representan vectores de numeros enteros y nos devuelve su producto punto.

```
Practica2Completa> productoPunto3D (1,4,6) (91,12,3)
157

Practica2Completa> productoPunto3D (4,4,2) (9,6,7)
74
```

productoPunto

Crea una funcion llamada **productoPunto** que recibe dos listas de enteros de longitud **n,m** que representan vectores de numeros enteros y nos devuelve su producto punto. Si no son de la misma longitud manda un: error "Necesitamos listas del mismo largo"

```
*Practica2Completa> productoPunto [4,4,2, 12, 2] [9,6,7]
*** Exception: Necesitamos listas del mismo largo
CallStack (from HasCallStack):
    error, called at Practica2Completa.hs:77:28 in main:Practica2Completa

Practica2Completa> productoPunto [4,4,2, 12, 2] [9,6,7,1,4]
94
```

productoCruz

Crea una funcion llamada **productoCruz** que toma 2 tripletas de enteros que representan dos vectores y nos devuelve su producto cruz.

$$R = a \times b = \begin{vmatrix} i & j & k \\ a_x & a_y & a_z \\ b_x & b_y & b_z \end{vmatrix} = \begin{vmatrix} a_y & a_z \\ b_y & b_z \end{vmatrix} i - \begin{vmatrix} a_x & a_z \\ b_x & b_z \end{vmatrix} j + \begin{vmatrix} a_x & a_y \\ b_x & b_y \end{vmatrix} k =$$

$$= (a_Y b_z - a_Z b_y)\,\mathbf{i} - (a_x b_z - a_Z b_x)\,\mathbf{j} + (a_x b_y - a_Y b_x)\,\mathbf{k}$$

```
Practica2Completa> productoCruz (1,4,6) (9,5,3)
(-18,51,-31)

Practica2Completa> productoCruz (6,4,3) (2,5,7)
(13,-36,22)
```

restaVectores

Crea una funcion llamada **restaVectores** que recibe 2 tripletas de enteros n, m que representan vectores y regresa: n - m

```
Practica2Completa> restaVectores (6,4,3) (2,5,7)
(4,-1,-4)

Practica2Completa> restaVectores (0,0,0) (2,5,7)
(-2,-5,-7)
```

normalAlTriangulo.

Crea una funcion llamada **normalAlTriangulo** que recibe 3 tripletas de enteros que representan los puntos de un triangulo (puedes suponer que te dan tres puntos no colineales y que el triangulo realmente se puede formar). Regresa un vector ortogonal a la superficie del triangulo.

```
ractica2Completa> normalATriangulo  (1,4,6) (4,2,5) (7,2,8)
(-6,-12,6)

Practica2Completa> normalATriangulo  (3,7,2) (1,3,2) (2,5,4)
(-8,4,0)
```

divisoresPropios

Crea la funcion **divisoresPropios** que recibe un entero y regresa una lista con sus divisores propios.

Un divisor propio de un nÃºmero es cualquier divisor que no es el mismo nÃºmero que el que divide.

```
Practica2Completa> divisoresPropios 21
[1,3,7]

Practica2Completa> divisoresPropios 42
[1,2,3,6,7,14,21]
```

sumaPares

Crea una funcion **sumaPares** que sume los primeros n numeros pares.

```
Practica2Completa> sumaPares 10
110

Practica2Completa> sumaPares 32
1056
```

primo

Crea una funcion **primo** que nos obtenga el n-esimo numero primo.

```
Practica2Completa> primo 8
19

Practica2Completa> primo 17
59
```

sumaDigitos

Crea la funcion **sumaDigitos** que recibe un entero n y suma sus digitos . No tiene que devolver un numero de un solo digito, puede que la suma de mas de 9.

```
Practica2Completa> sumaDigitos 17
8

ractica2Completa> sumaDigitos 321579
27
```

imprimeUnEntero

Crea una funcion llamada **imprimeUnEntero** que recibe un entero (puede ser positivo o negativo) y regresa su representacion en String. Obviamente no puedes usar la funcion *show*.

```
imprimeUnEntero (-1234)
"-1234"

Practica2Completa> imprimeUnEntero 98412
"98412"
```

Tests.

Como se habran dado cuenta, tambien agregamos un archivo de pruebas. Para poder correrlo solo deben de cargarlo con

```
> ghci Practica2Tests.hs
```

Y para correr un test en especifico:

```
Practica2Tests> quickCheck nombre_del_test
```

Por ejemplo:

```
Practica2Tests> quickCheck prop_esPrimo
+++ OK, passed 100 tests.
```

Si sale un error, les marcara el caso que probo y que resulto en un error

```
*Practica2Tests> quickCheck prop_imprimeUnEntero
*** Failed! Falsifiable (after 1 test):
0
```

Aqui nos dice que el test prop_imprimeUnEntero fallo porque al pasarle "0" el resultado no fue el que esperaba.

Para correr todos los tests solo escriben runTests

```
*Practica2Tests> runTests
=== prop_divisoresPropios from Practica2Tests.hs:7 ===
+++ OK, passed 100 tests; 11 discarded.

=== prop_sumaPares from Practica2Tests.hs:12 ===
+++ OK, passed 100 tests; 22 discarded.

=== prop_esPrimo from Practica2Tests.hs:15 ===
+++ OK, passed 100 tests.

=== pzrop_sumaDigitos from Practica2Tests.hs:20 ===
+++ OK, passed 100 tests.

=== prop_imprimeUnEntero from Practica2Tests.hs:25 ===
+++ OK, passed 100 tests.

=== prop_productoPunto3D_conmutativa from Practica2Tests.hs:28 ===
+++ OK, passed 100 tests.

=== prop_productoPunto from Practica2Tests.hs:31 ===
+++ OK, passed 100 tests; 19 discarded.

=== prop_productoPunto_conmutativa from Practica2Tests.hs:34 ===
+++ OK, passed 100 tests.
```

```
=== prop_productoCruz_cancelacion from Practica2Tests.hs:37 ===
+++ OK, passed 100 tests.

=== prop_restaVectores from Practica2Tests.hs:40 ===
+++ OK, passed 100 tests.

=== prop_restaVectores_ceros from Practica2Tests.hs:46 ===
+++ OK, passed 100 tests.

=== prop_normalATriangulo from Practica2Tests.hs:49 ===
+++ OK, passed 100 tests; 10 discarded.

True
```

Instalando QuickCheck

Es probable que falte instalar la biblioteca de QuickCheck (la que se encarga de los tests) en su computadora. Primero revisen si tienen cabal instalado (que deberian de tenerlo) con el comando

```
> cabal --version
cabal-install version 2.4.0.0
compiled using version 2.4.0.1 of the Cabal library

> cabal install QuickCheck
abal install QuickCheck
Warning: The install command is a part of the legacy v1 style of cabal usage.

Please switch to using either the new project style and the new-install
command or the legacy v1-install alias as new-style projects will become the
default in the next version of cabal-install. Please file a bug if you cannot
replicate a working v1- use case with the new-style commands.

For more information, see: https://wiki.haskell.org/Cabal/NewBuild

Warning: /home/jason/.cabal/config: Unrecognized stanza on line 140
/home/jason/.cabal/config: Unrecognized field installdir on line 117
Warning: The package list for 'hackage.haskell.org' is 89 days old.
Run 'cabal update' to get the latest list of available packages.
Resolving dependencies...
Downloading  splitmix-0.1.0.4
Downloaded   splitmix-0.1.0.4
Starting     splitmix-0.1.0.4
Building     splitmix-0.1.0.4
Completed    splitmix-0.1.0.4
Downloading  QuickCheck-2.14.2
Downloaded   QuickCheck-2.14.2
Starting     QuickCheck-2.14.2
Building     QuickCheck-2.14.2
Completed    QuickCheck-2.14.2
```

windows:

Los comandos son los mismos, solo deben ejecutarlo desde la powershell.