TALLER 3

Trabajo puede ser entregado de manera individual. Favor enviar sus programas adjuntos en una carpeta.

1. Escriba un programa que imprima una lista de números del 2 al 20, y que muestra si el número es divisible por 2, por 3, por ambos o por ninguno. La salida en pantalla se puede ver algo así

```
Num   Div by 2 and/or 3?
---   ------------------
2            by 2
3            by 3
4            by 2
5            neither
6            both
7            neither
```

2. Escriba un programa que repetidamente le pida al usuario tres valores (reales) a, b, c, para la ecuación cuadrática

$$a * x^2 + b * x + c = 0,$$

   Usando la famosa formula para las raíces de este problema, el programa debe reconocer si hay raíces reales y calcular estos valores. Como resultado, muestre el número de raíces reales y sus valores. El programa debe parar si todos los valores son 0.

3. Modifique el programa gfc.py para calcular el mínimo común múltiplo (least common multiple en inglés) de dos números enteros.

4. Python incluye un modulo para la generación de números aleatorios (llamado random). Abajo se muestra un ejemplo que imprime en pantalla 20 valores aleatorios entre 0 y 1.

```
# random_example.py
# Generate a short list of random number between 0 and 1

import random

for i in range(20):
   a = random.random()
   print (a)
```

   Escriba un programa simple de Python para generar 10000 números aleatorios entre 0 y 1. Realice un test de que tan aleatorio es el generador de números, contando el número de veces que el número cae dentro de 10 rangos distintos entre 0 y 1 (por ejemplo, 0 a 0.1, 0.1 a 0.2, etc.). Imprima el número total de veces para cada rango en la pantalla. Este es un ejemplo de cómo se puede ver la salida del programa:

```
0 1009
1 1048
2 1001
3 1038
4 1008
5 959
6 993
7 925
8 1017
9 1002
```

AYUDA: Cree un arreglo de números enteros (mire el modulo `numpy`) con 10 elementos e inicie con ceros. Para cada número aleatorio, sume una unidad al rango apropiado y así poder adicionar todos los números.

NOTA: En C o Fortran cada vez que corra el programa, Ud. obtendrá siempre la misma serie de números aleatorios. Esto no es muy buena idea, pero el código para obtener series de números aleatorios distintas cada vez que se corre el programa es complicado. Python fija la semilla del número aleatorio (*seed*) con el reloj del computador, así que no debe ser un problema.