Computación científica usando software libre
Lista de ejercicios 1. Entrega 30 de marzo

1. (a) Hacer una lista de 5 programas de software libre que hayas usado.
   (b) Para cada uno, nombrar un programa comercial correspondiente, si existe
   (c) Listar algunas de las ventajas y desventajas del programa libre con respecto al programa comercial.

2. Usar Sage para calcular la cantidad de enteros positivos hasta un millon que dejan resto 3 al dividir por 7 (sugerencia, dado n, el resto de dividir entre 7 se obtiene usando `n % 7` en Python.)

3. Escribir una línea en Sage que tome por lo menos dos segundos en evaluarse.

4. (a) Crear un ejemplo de una *lista* de python.
   (b) Crear un ejemplo de una *tupla* de python.
   (c) Crear un ejemplo de un *diccionario* de python.
   (d) Crear un ejemplo de una *cadena* de python.
   (e) Crear un ejemplo de un *conjunto* de python.

5. Deteriminar cuáles de los siguientes objetos de Python $a, b, c, d, e, f, g, h, i, j$ son *inmutables* (i.e. no pueden ser cambiados desde Python):

```
a = 'Hola, mundo!'
b = int(123456789)
c = 2/3
d = [1,2,3,4,5]
e = (1,2,3,4,5)
f = (1,2,3,4,[5])
g = {1:'a', 2:'b'}
h = set([1,2,3])
class i: pass
j = i()
```

6. Considerar la implementación recursiva de la función factorial:

```
def factorial(n):
    if n <= 2: return n
    return n * factorial(n-1)
```

   (a) Probar que funciona para los primeros valores de n, y estudiar el tiempo que demora en la práctica (sugerencia: utilizar la función `timeit`).
   (b) ¿Cuál es el mayor valor de n para el que esta implementación funciona? ¿Qué es lo que falla?

Computación científica usando software libre
Lista de ejercicios 2. Entrega 13 de abril

1. Encontrar un *bug* en Sage. Intentar averiguar si el bug es conocido.
2. Leer el ensayo *Mathematica viene al CIMAT*:
   `http://sums.math.mcgill.ca/~jordi/rants/mathematica.html`
3. El código que sigue tiene un *bug*. Intentar

   ```
   def f(a, L=[]):
       L.append(a)
       return L

   print f(1)
   print f(2)
   print f(3)
   ```

   Esto imprime

   ```
   [1]
   [1, 2]
   [1, 2, 3]
   ```

   pero lo que queremos que imprima es

   ```
   [1]
   [2]
   [3]
   ```

   Solucionar este problema. (Sugerencia: buscar la respuesta en el tutorial de Python).
4. Escribir una función rápida en Cython que determine si un entero positivo (a lo sumo 2^{31}) es una potencia perfecta de 3. Comparar la velocidad con la de una función de Python que haga lo mismo.
5. (a) Crear una clase de python `hlist` que deriva de la clase `list`, pero agrega un nuevo método `__hash__`:

   ```
   def __hash__(self):
       ???
   ```

   (b) ¿Pueden crearse diccionarios con objetos de tipo `hlist` como índices? Dar un ejemplo.

   (c) Si la respuesta en (b) es afirmativa, mostrar algo que no está bien como resultado. Es decir, mostrar *por qué* es una mala idea definir un método hash para objetos mutables.
6. Dar *tres* ideas para proyectos que podrían interesarte como posible proyecto para este curso.

Computación científica usando software libre
Lista de ejercicios 3. Entrega 15 de mayo

1. Dar un ejemplo que muestre que el "cuerpo" `CC` de números complejos de 53 bits de precisión *no* es un cuerpo en el sentido matemático estricto. Para esto, mostrar que por lo menos uno de los axiomas de cuerpo fallan.

2. (a) Encontrar un número probablemente primo con exactamente 2009 cifras decimales, utilizando el comando `next_probable_prime` de Sage.
   (b) Estimar (justificando) el tiempo que demoraría en Sage *probar* que el primo probable de la parte anterior es efectivamente primo, utilizando el comando `is_prime`.

3. Considerar la secuencia $p_1, p_2, \ldots$, donde $p_1 = 2$, $p_2 = 3$, $p_3 = 5$, etc. de todos los números primos $\leq 10^6$.
   (a) ¿Para cuantos n se cumple $p_{n+1} = p_n + 2$?
   (b) ¿Para cuantos n se cumple $p_{n+1} = p_n + 4$?
   (c) ¿Para cuantos n se cumple $p_{n+1} = p_n + 6$?
   (d) Enunciar algunas conjeturas basadas en estos cálculos.

4. Crear la matriz 10×10 cuyas entradas son los primeros 100 números primos, i.e.

$$A = \begin{pmatrix} 2 & 3 & 5 & 7 & 11 & 13 & 17 & 19 & 23 & 29 \\ 31 & 37 & 41 & 43 & 47 & 53 & 59 & 61 & 67 & 71 \\ 73 & 79 & 83 & 89 & 97 & 101 & 103 & 107 & 109 & 113 \\ 127 & 131 & 137 & 139 & 149 & 151 & 157 & 163 & 167 & 173 \\ 179 & 181 & 191 & 193 & 197 & 199 & 211 & 223 & 227 & 229 \\ 233 & 239 & 241 & 251 & 257 & 263 & 269 & 271 & 277 & 281 \\ 283 & 293 & 307 & 311 & 313 & 317 & 331 & 337 & 347 & 349 \\ 353 & 359 & 367 & 373 & 379 & 383 & 389 & 397 & 401 & 409 \\ 419 & 421 & 431 & 433 & 439 & 443 & 449 & 457 & 461 & 463 \\ 467 & 479 & 487 & 491 & 499 & 503 & 509 & 521 & 523 & 541 \end{pmatrix}$$

   Sea v el vector cuyas entradas son los primeros 10 números primos.
   (a) ¿Cuál es el mínimo común múltiplo de los denominadores de las entradas del único vector x con $Ax = v$?
   (b) Repetir el experimento cambiando 10 por 100, es decir con A una matriz de 100×100.

5. (a) Calcular la suma s de las potencias centésimas de los enteros positivos hasta 10^7 (diez millones). Para la respuesta, dar solamente el número de cifras decimales de s.
   (b) Evaluar la suma infinita
$$\sum_{i=1}^{\infty} \frac{1}{n^{100}}.$$