Segundo Parcial

4 de abril de 2020

Indicaciones generales

- Este es un examen **individual** con una duración de **120 minutos: de 8:00 a 10:00**.
- En `e-aulas` puede acceder a las diapositivas y a la sección correspondiente a este parcial.
- Solamente será posible tener acceso a `e-aulas.urosario.edu.co` y a los sitios web correspondientes a la documentación de Python dispuestos por el profesor.
- Celulares y otros dispositivos electrónicos no deben ser utilizados durante el examen.
- El estudiante solo podrá disponer de hojas en blanco como borrador de apuntes (opcional).
- El estudiante puede tener una hoja manuscrita de resumen (opcional). Esta hoja debe estar marcada con nombre completo.
- Como este parcial se realizará virtualmente, solo se considerarán las respuestas al parcial de aquellos estudiantes que se conecten a través de la plataforma, enciendan su cámara y micrófono durante la realización del parcial, y suban el parcial a e-aulas antes de retirarse del aula virtual.
- La actividad en `e-aulas` se cerrará a la hora en punto acordada. La solución de la actividad debe ser subida antes de esta hora. El material entregado a través de `e-aulas` será calificado tal como está. Si ningún tipo de material es entregado por este medio, la nota de la evaluación será 0.0.
  Se aconseja subir a e-aulas versiones parciales de la solución a la actividad.
- **Cualquier incumplimiento de lo anterior conlleva la anulación del examen.**
- Las respuestas deben estar totalmente justificadas.
- Use las plantillas `plantillaPY.Z`, con `Y = 1,2,3` y `Z = py`, para dar su solución.
- **Entrega**: tres archivos con extensión `'.py'` (Python) según el caso, conteniendo el código solución, basado en las plantillas. Nombre los archivos como `pY.Z`, con `Y = 1,2,3` y `Z = py, txt`.
  Importante: no use acentos ni deje espacios en los nombres de los archivos que cree.

1. [20 ptos.] Implemente la función `char_to_ascci`, que recibe una cadena de caracteres con caracteres alfanuméricos. La función debe retornar una cadena en donde los caracteres no numéricos han sido reemplazados por su código ASCCI. Para transformar de caracter a código ASCCI utilice el método `ord()`. Por ejemplo, la cadena `"aB1c5"` produce la salida `"97661995"`.

2. [40 ptos.] El Instituto Nacional de Salud (INS)[1] posee los datos históricos de casos confirmados de covid-19 para todo el terroritorio nacional de los primeros 20 días en los que no hubo cuarentena, tal como se muestra en la Figura 1.

   El INS ha pedido nuestra colaboración en los siguientes aspectos:

   a) Generar una funcion llamada *prediction_covid*(d) que calcule y retorne la predicción de número de casos de infección según el número del día desde que apareció el primer caso. Se sabe que la función que mejor describe la aparición de nuevos casos es $f(d) = 10 * e^{(0.23*d)}$,

[1] https://www.ins.gov.co/

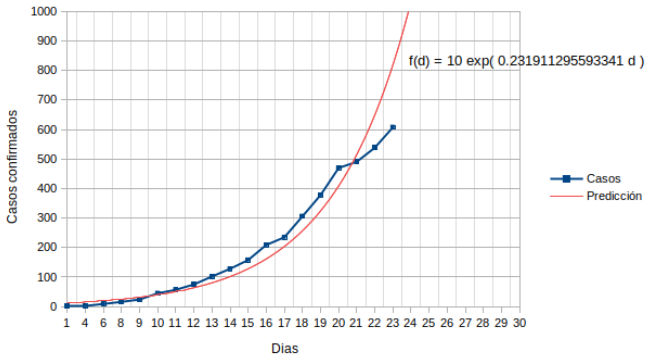

Figura 1: Predicción de casos covid por días para Colombia

donde: d es el número de días desde la aparición del primer caso de infección y $f(d)$ es la predicción de número de casos de infección de covid-19. Esta función será invocada para saber los casos de infección a los 28, 29, 30 y 31 días después de la aparicion del primer caso de infección.

b) Generar una función llamada $generating\_dictionary(s, d, y)$ que almacene en un diccionario s cada uno de los valores de d como llave y y como valor de dicha llave. Esta función no retorna nada. Esta función será utilizada para almacenar en un diccionario s los valores de dıas d y número de casos y calculados en el punto anterior.

c) Generar una función llamada $store\_at\_disk(s)$ que guarde en un archivo database.txt los datos del diccionario s del punto anterior obteniendo un resultado como el del Cuadro 1. Esta función no retorna nada.

Database with covid predictions for Colombia	
Día	Casos
28	6264
29	7884
30	9923
31	12489

Cuadro 1: Database with covid predictions for Colombia

En su archivo de texto no cree los bordes (lineas de separación) de la tabla. Utilice la plantilla p2.py para resolver este ejercicio. No tiene que entregar en e-aulas el archivo database.txt dado que este deberá ser generado en el momento de la ejecución por parte del profesor evaluador

3. [40 ptos.] Implemente la clase `NumeroComplejo`, la cual representa un número de la forma $a + bi$. Esta clase debe tener el atributo a que representa la parte real del número complejo y el atributo b que representa la parte imaginaria número complejo.

a) Implemente el constructor de la clase que permita recibir como parámetros los valores de la parte real y la parte imaginaria del número complejo.

b) Implemente el método `modulo` que retorna el módulo del número complejo. Sea $z = a+bi$, el módulo de z se denota como

$$|z| = \sqrt{a^2 + b^2} \tag{1}$$

c) Implemente el método `argumento` que retorna el argumento del número complejo. Sea $z = a + bi$, el argumento de z se denota como

$$arg(z) = \begin{cases} \arctan \frac{b}{a} & a \neq 0 \\ 90 & a = 0 \end{cases} \tag{2}$$

d) Implemente el método `__str__` que retorna una versión en string del número complejo de la forma: $a + bi$

Utilice la plantilla `NumeroComplejo.py` para crear los metodos de la clase. Utilice el archivo de prueba `pruebaNumeroComplejo.py` para validar los resultados.

Nota: La función `atan` del modulo `math` calcula el Arcotangente en radianes. Para convertir radianes a grados multiplique por `180` y divida por `pi`.