



SEGUNDO PARCIAL

13 de octubre de 2020

Indicaciones generales

- Este es un examen **individual** con una duración de **110 minutos: de 3:00 pm a 4:50 pm**.
- 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.py**, con $Y = 1, 2$, para dar su solución.
- **Entrega:** dos archivos con extensión **‘.py’** (Python) según el caso, conteniendo el código solución, basado en las plantillas. Nombre los archivos como **pY.py**, con $Y = 1, 2$.

Importante: no use acentos ni deje espacios en los nombres de los archivos que cree.

1. Escriba una función en Python llamada **reduccion(listaA)** que reciba como parámetro de entrada una lista **listaA** de longitud indeterminada, la cual está conformada por números del 0 al 9 y letras de la **a** a la **z**. Esta función debe retornar una tupla conformada por los siguientes elementos:

- a) [20 ptos.] Una lista que contenga las letras encontradas en **listaA**
- b) [20 ptos.] La suma de los números dentro de **listaA**
- c) [20 ptos.] La multiplicación de los números dentro de **listaA**

Nota: Debe utilizar *While* para recorrer la lista.

Un ejemplo de la ejecución de la función **reduccion** se muestra a continuación:

```
1 | test(reduccion(["a",2,1,"s",2,"a"])) == (["a", "s", "a"], 5, 4)
2 | test(reduccion([2,"h","r",3,"s",2,"a"])) == (["a", "s", "r", "h"], 7, 12)
3 | test(reduccion([8,"d",4,"s",9,"z"])) == (["z", "s", "d"], 21, 288)
4 | test(reduccion([1,"g","a",8,"t",8,"c"])) == (["c", "t", "a", "g"], 17, 64)
```



Utilice la plantilla adjunta para la realización del ejercicio.

2. [40 ptos.] Escriba la función `sumas(cad)`, que recibe una cadena de caracteres compuesta de caracteres numéricos y espacios, por ejemplo:

```
1 || "132_111_54_7"
```

y retorna un diccionario con claves enteras que van de 0 al número de subcadenas separadas por espacios que contenga la cadena. Como valores el diccionario contiene la suma de los dígitos presentes en cada subcadena. Por ejemplo, para la cadena presentada el diccionario retornado sería:

```
1 || {0:6, 1:3, 2:9, 3:7}
```