

Turno 2

Preparación y consignas generales:

- Para comenzar, cree un proyecto en PyCharm y dentro del mismo cree un archivo **.py** cuyo nombre tenga el formato **legajo-apellido** (donde "legajo" es su número de legajo y "apellido" es su apellido).
- En la misma carpeta de ese proyecto, descargue y guarde el archivo **"entrada.txt"** que se provee con este enunciado.
- Se le pedirá que procese una cadena de caracteres cuya carga debe hacerse **obligatoriamente** desde el archivo de texto **"entrada.txt"** según técnicas que fueron explicadas en fichas y en clases prácticas.
- El texto que cargue desde ese archivo finaliza con "." y cada palabra de ese texto está separada de las demás por un (y solo un) espacio en blanco. No hay saltos de línea en el archivo.
- El programa **debe** incluir una función principal para lanzar el programa desde el script principal.
- El programa **debe** tener control de ejecución del script principal con la variable **__name__**.
- El programa **debe** tener **al menos** una función simple desarrollada por el estudiante con parámetros y con retorno de resultados.
- El programa **debe** procesar el texto caracter a caracter (a razón de uno por vuelta de ciclo, **con un único ciclo** para todo el proceso).
- El programa que entreguen **NO debe** usar un menú de opciones ni ningún tipo de carga por teclado en ninguna parte del programa por ninguna razón. El texto **debe** ser levantado estrictamente desde el archivo **"entrada.txt"**.
- La secuencia y el formato de las instrucciones de salida por pantalla de su programa, debe ser **obligatoriamente y tal cual** la que se indica a continuación. **No cambie los mensajes, no cambie los caracteres en cada mensaje, no cambie la forma de cada mensaje, ni cambie el orden de cada print()**:

```
print("Primer resultado:", r1)
print("Segundo resultado:", r2)
print("Tercer resultado:", r3)
print("Cuarto resultado:", r4)
```
- Copie y pegue las cuatro instrucciones anteriores **tal como están** al final de su función principal. No agregue **ninguna** otra llamada a **print()** en **ninguna** parte de su programa. Guarde los resultados que vaya calculando en las variables r1 (para el primer resultado pedido), y en r2, r3 y r4 respectivamente (para los resultados que siguen).

Enunciado:

Se pide desarrollar un programa en Python que permita procesar un texto completo contenido en una variable de tipo cadena de caracteres (cargado desde el archivo "entrada.txt" de acuerdo a todo lo expresado en la sección anterior), que haga lo siguiente:

- Determinar la cantidad de palabras que tienen un dígito en la segunda o en la tercera posición y además incluyen dos o más consonantes pero a partir de la cuarta posición, incluida la cuarta (en minúscula o mayúscula). Por ejemplo, en el texto: **"Ax1efDa** es una password aceptable pero **12r3aei** no lo es." Respuesta: hay solo una palabra que cumple: **"Ax1efDa"**. La palabra **"123aei"** tiene los dígitos pedidos pero no tiene ninguna consonante a partir de la cuarta posición, por lo que no cuenta.
- Determinar el **porcentaje entero** de palabras (con respecto al total de palabras del texto), de las palabras que tienen al menos una vocal (en minúscula o mayúscula) y finalizan con un dígito. Por ejemplo, en el texto: **"Con los procesos pr1ae8 y gri78x9** no era posible pero lo es a partir del trA2j." hay dos palabras que cumplen el criterio: **"pr1ae8"** y **"gri78x9"**. Como hay 16 palabras en total, el **porcentaje entero** pedido es

del 12 por ciento (aclaración: en el cálculo del porcentaje haga primero la multiplicación y luego la división).

3. Determinar cuántas palabras tienen cuatro caracteres o más pero en las primeras tres posiciones solo tienen vocales (en minúscula o mayúscula, y sin importar si luego aparecen otras vocales más atrás). Por ejemplo, en el texto "IEEE es una sigla conocida y también ACMLibrary pero no AINews." hay solo una palabra que cumple: "IEEE". La palabra "ACMLibrary" tiene la longitud correcta pero no tiene tres vocales en las primeras tres posiciones, y "AINews" también tiene longitud correcta pero tiene solo dos vocales en las tres primeras posiciones.
4. Determinar cuántas palabras incluyen la expresión "de" (con cualquiera de sus letras en minúscula o mayúscula) pero de tal forma que después de ella aparezca una "t" (minúscula o mayúscula) en cualquier lugar. Por ejemplo, en el texto: "Desde el **delantal** y las ataderas adiestradas se ve el **adelanto** de las dietas saludables por lo que **atender**te con el especialista de apellido **Deutedet** es lo mejor." hay cuatro palabras que cumplen: "**delantal**", "**adelanto**", "**atender**te" y "**Deutedet**." La palabra "Desde" no cuenta porque no contiene una "t" luego de la expresión "De", "ataderas" tampoco cuenta porque tiene una "t" pero está antes de la expresión "de", y "adiestradas" y "dietas" no cuentan porque no tienen la expresión "de".

Criterios generales de evaluación.

- Planteo sin carga por teclado en ninguna parte: 0 puntos (**reprueba si no cumple**).
- Instrucciones de salida tal cual se indicó: 0 puntos (**reprueba si no cumple**).
- Nombre del archivo fuente correcto: 0 puntos (-1 si no cumple).
- Apertura correcta del archivo "entrada.txt": 0 puntos (-1 si no cumple).
- Planteo en base a un único ciclo: máximo 0 puntos (-2 si no cumple).
- Inclusión correcta de una función principal: **máximo 2 puntos**.
- Inclusión correcta de al menos una función con parámetros y retorno: **máximo 2 puntos**.
- Control correcto de ejecución del script principal: **máximo 1 punto**.
- Resultado correcto del ítem 1: **máximo: 3 puntos**.
- Resultado correcto del ítem 2: **máximo: 4 puntos**.
- Resultado correcto del ítem 3: **máximo: 5 puntos**.
- Resultado correcto del ítem 4: **máximo 6 puntos**.
- Para aprobar el parcial, el estudiante debe llegar a un mínimo de alrededor de 12 puntos (un porcentaje de al menos 55% del puntaje máximo de 22 que puede ser alcanzado).