



UNIVERSIDAD ABIERTA INTERAMERICANA
FACULTAD DE TECNOLOGÍA INFORMÁTICA
MATERIA: PROGRAMACIÓN II
2º Parcial
Teórico y Práctico

Alumno:

Fecha:

Comisión – Localización – Turno:

Práctica:

Teoría:

Nota:

Temas para evaluar: Funciones de orden superior, Expresiones regulares (Regex), Módulos más utilizados en Python.

Objetivos:

Comprender las pautas del proceso para la creación de software desde la perspectiva de la Ing. SW.
Comprender cómo se aplican las técnicas del paradigma estructurado y POO en el lenguaje python

Modalidad: Parcial domiciliario

Requisitos para aprobar: Para que el parcial esté aprobado el alumno deberá tener correctamente desarrolladas el 60% de la teoría y resuelto el ejercicio práctico.

Tiempo:

Recomendaciones:

- a) Lea todo el parcial antes de comenzar a responder.
- b) Desarrolle una redacción clara y precisa contestando lo que la pregunta requiere.
- c) Observe la ortografía ya que la misma es parte del parcial.
- d) Si considera que no comprende alguna consigna antes de comenzar consulte a su profesor.

Parte Práctica:

Realizar la resolución del siguiente enunciado en Python:

1. Calcular las siguientes métricas en base al texto a continuación:

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. (40)

- Cantidad de veces que aparece cada palabra.
- Longitud de cada palabra
- Tener cuidado con la puntuación del texto, la misma **no** debe considerarse en las palabras.

Utilizar **únicamente** la combinación de funciones de orden superior y diversos tipos de colecciones

2. Dada la siguiente lista de caracteres correspondientes a una secuencia de ADN:

```
TGCCACAAATGTGACAGGACGCCGATGGGTACCGACTTTAGGTCGAGCACAGTTCGGTAACGGA
GAGACCCTGCGGCGTACTTCATTATGTATATGGAACGTGCCAAGTGACGCCAGGCAAGTCTCAGC
TGGTTCCTGTGTTAGCTCGAGGGTAGACATACGAGCTGATTGAACATGGGTTGGGGGCCTCGAACC
GTCGAGGACCCCATAGTACCTCGGAGACCAAGTAGGGCAGCCTATAGTTTGAAGCAGAACTATTTCC
GGGGGCGAGCCCTCATCGTCTCTTCTGCGGATGACTCAACACGCTAGGGACGTGAAGTCGATTCC
TTCGATGGTTATAAATCAAAGACCGAGGACCGCAAGGTGCGGCGGTGCACAAGCAATTGACAATA
ACCACCGTGATTTCATTATGGTACCAGGAACTTTAAGCCGAGTCAATGAAGCTCGCATTACAGTGTTT
ACCGCATCTTGCCGTACT. (35)
```

- Encontrar la cantidad de veces que aparece la serie "CATT"
- Probar que existe la serie "CAAAGACCGAGGACC"
- Probar que NO existe la serie "TTGAACCAA"

Utilizar **únicamente** expresiones regulares. Es válido el uso de cualquiera de los métodos del módulo re.

3. Definir la clase alumno con los atributos Nombre, Edad, Carrera y DNI. Generar una lista de alumnos y mediante la utilización de funciones de orden superior y mapear dicha lista a la clase mostrando el resultado por consola. (25)
- Recordar utilizar docstrings para la documentación de las funciones desarrolladas.
 - Respetar las buenas prácticas de la codificación.
 - Utilizar todos los artefactos que crea conveniente así como built-in functions o librerías particulares. Se valorará especialmente la creatividad a la hora de resolver la problemática como también la impronta personal expuesta.