



Analizando H5N1 (SECCION 1)

El Centro de Operaciones Nacional de la Emergencia de Influenza Aviar pone a disposición de la ciudadanía la base de datos de esta emergencia con toda la información relevante de los muestreos realizados por el SAG, con el objetivo de entregar de la forma más completa, detallada, transparente y actualizada los datos, región por región, de esta contingencia nacional. Estos datos se encuentran en un archivo de texto.

Objetivo

En este proyecto usted debe escribir un programa en Python 3, que dé respuesta a lo siguiente:

- a. Que indique la cantidad de casos de secuenciación por región de H5N1.
- b. Que indique la cantidad de casos negativos para el mes de abril del año 2023.
- c. Que indique la cantidad de casos negativos de la especie “Yunco”.
- d. Que indique la cantidad de casos negativos para la especie “Lile” para el mes de junio 2023.
- e. Que grafique la cantidad de casos negativos (incidencias) de las siguientes especies: Gaviota, Piquero, Salteador, Pelicano y Guanay, considerando un gráfico de barras.

Entrada

Los datos se encuentran en un archivo llamado “protocolo_vigilancia.txt”, que se encuentra disponible para su descarga en la plataforma UCMVirtual del curso. Cada línea contiene la siguiente información: número de formulario, motivo de la visita, fecha de la muestra, región, comuna, especie, número de animales muestreados, cantidad de muestras, secuencia y clasificación. Cada dato en la línea viene separado por una coma.

Salida

La salida del programa deberá quedar registrada en un archivo de texto llamado “resultadoS3.txt”, con el siguiente formato:

```
Autor(es): nnnnnnn - nnnnnnn  
  
Cantidad de casos de secuenciación por región:  
  
    Nombre de la región: xxxx  
  
    Nombre de la región: xxxx  
  
    ...  
  
Casos negativos mes abril del año 2023: xxxx  
  
Casos negativo especie Yunco: xxxx  
  
Incidencias 06/2023 del “Lile” : NNN.
```

Mostrar el grafico solicitado.

Rúbrica de Evaluación

	1	3	5	7	Total
Revisión del Código. (20%)	Código desordenado, sin uso de las normas de programación, no utiliza comentarios, nombres de variables inadecuados.	Código Ordenado, usa indentación. Usa identificadores de variables adecuado. Incorpora comentarios mínimos o existe sobresaturación de comentarios.	Código ordenado. Uso adecuado de variables (listas, string, etc.). Uso adecuado de estructuras algorítmicas Uso eficiente de los comentarios.	Cumple con todas las normas de programación.	
Solución Computacional. (40%)	No se ejecuta. Errores de Ejecución. Programa cae en Loop.	Ejecuta pero con errores de lógica. Ejecuta pero no llega a la solución.	Llega a una solución parcial. Tarda más de un minuto en llegar a la solución. No cumple con el formato de salida.	Llega a la solución en menos de un minuto cumpliendo exactamente con el formato de salida.	
Interrogación (40%)	No da respuesta a lo preguntado entrega respuesta errónea.			Da respuesta correcta a lo consultado	

Requerimiento base:

Como mínimo su programa debe presentar la estructura siguiente.

```
def lectura_datos(...): ...

def funcion_a(...): ...

def funcion_b(...): ...

def funcion_c(...): ...

def funcion_d(...): ...

def funcion_e(...): ...

def genera_salida(...): ...

if __name__ == "__main__": ...
    i? = funcion_a(i?) # Respuesta al punto a de la tarea
    i? = funcion_b(i?) # Respuesta al punto b de la tarea
    i? = funcion_c(i?) # Respuesta al punto c de la tarea
    i? = funcion_d(i?) # Respuesta al punto d de la tarea
    i? = funcion_e(i?) # Respuesta al punto e de la tarea
    genera_salida(i?)
```

Debe respetar estrictamente el nombre de las funciones, usted completar lo necesario en cada función.

INDICACIONES:

Debe ser desarrollado por exactamente por dos estudiantes (ni más ni menos), por lo tanto con fecha 13 de junio deben subir a la plataforma UCMvirtual un archivo word conteniendo la identificación de cada uno de los integrantes.

La fecha de entrega del proyecto es a través de la misma plataforma es el día viernes 21 de Junio hasta las 23:30 hrs.