



503203/503201/503215 Programación Programación Usando Cadenas y Listas

EQUIPO COORDINACIÓN DE PROGRAMACIÓN

16 de junio de 2022

- 1.- La *discalculia* o *dislexia de los números* es una disfunción del aprendizaje que afecta la capacidad de un niño para comprender, aprender y realizar operaciones matemáticas y basadas en números. Esta afecta a entre el 5 % y el 7 % de la población infantil en edad escolar primaria y una de las evidencias de que un/a niño/a la sufra es que tendrá dificultad para conectar los símbolos numéricos (p.e., 5) con las palabras correspondientes (p.e., cinco). Para observar si esta evidencia aparece en algún niño se le ha pedido que escriba un programa en Python que lea 5 números desde teclado y que el programa escriba uno de ellos (elegido al azar) en palabras. Con esto, la prueba para los niños será indicar a qué número corresponde el texto.

Entradas: La entrada al programa estará compuesta por 5 números. Los números deben ser mayores o iguales a cero y menores a 100.

Salidas: La salida del programa será un string señalando uno de los números en palabra.

Ejemplo de entrada 1: 67 5 41 96 4

Ejemplo de salida 1: cinco

Ejemplo de entrada 2: 41 22 75 19 87

Ejemplo de salida 1: ochenta y siete

- 2.- Hace tiempo que no llovía tanto. En eso coinciden los meteorólogos, pero se preguntan en qué momento hubo precipitaciones similares a la de los últimos 5 días en Concepción. La Dirección Meteorológica de Chile cuenta con registros diarios de las precipitaciones en Concepción desde hace 30 años y encontrar el evento más parecido al de nuestros días se trata de buscar en todos esos registros aquellos que cuenten con el promedio de las diferencias menores en relación con las precipitaciones de los últimos 5 días. Escriba un programa Python que, dado un evento de pluviometría de 5 días, determine el evento más parecido en el pasado.

Entradas: La entrada estará compuesta por un conjunto de entre 10 y 10000 enteros no negativos correspondientes a la cantidad de precipitaciones diarias medidas en Concepción durante los últimos años, los últimos 5 números corresponden a las precipitaciones de los últimos 5 días. Se debe ingresar la secuencia hasta que se ingrese -1 respetando que no sean menos de 10 datos ingresados ni más de 10000.

Salidas: La salida indicará primero dos valores, la posición de inicio y término del evento pluviométrico más parecido al actual de los últimos 5 días y a continuación la diferencia de promedio menor.

Ejemplo de entrada 1: 67, 5, 41, 96, 4, 20, 10, 0, 0, 0, 21, 32, 34,
41, 12, 15, 6, 3, 0, 0, 0, 0, 4, 8, 12, 21, 10, 25, -1

Ejemplo de salida 1: 6 10 9.8

Observación: El evento 6 al 10 corresponde a los registros 10, 0, 0, 0 y 21. La diferencia entre estos registros y los últimos 5 (8, 12, 21, 10 y 25) es $2+12+21+10+4=47$, lo que da un promedio de $47/5=9.8$.

3.- Un conjunto X de *expresiones de paréntesis correctamente construida* esta compuesto por varias cadenas de símbolos, todas formadas sólo por símbolos “(” y “)” y definida de la siguiente forma:

- un string vacío pertenece a X
- si A pertenece a X , entonces (A) pertenece a X
- si A y B pertenecen a X , entonces la concatenación AB pertenece a X

Por ejemplo, las siguientes representan *expresiones de paréntesis correctamente construidas* y, por tanto, pertenecen a X :

`()()`

`((()))`

mientras que las expresiones:

`(())()`

`()))()`

no pertenecen a X .

Escriba un programa en Python que lea expresiones de paréntesis y determine si son correctamente construidas.

Entradas: La única entrada al programa es un string compuesto sólo de paréntesis. La entrada puede incluir el string vacío.

Salidas: El programa debe indicar si la cadena ingresada es o no una *expresiones de paréntesis correctamente construidas*

Ejemplo de entrada 1: `((()))`

Ejemplo de salida 1: Expresión de paréntesis correctamente construida

4.- Un investigador tiene el código genético (ARN) del nuevo virus que producirá la pandemia del 2021, el cual ataca al genoma humano de una forma muy rara. Este virus es capaz de destruir cada parte del código genético de un humano que es igual a su código genético. Sabiendo eso, el investigador quiere crear nuevos tipos de bacterias que sean resistentes a este virus y en sus pruebas quiere determinar que % de preservación de su código genético original se conserva luego de que éstas son atacadas por el virus. Para ayudarlo en su investigación, escriba un programa en Python que reciba el código genético (ADN) de una bacteria y que calcule la preservación de ésta cuando es atacada por el virus. El código genético de la bacteria y del virus están compuestos sólo por **A**, **C**, **G** y **T**¹.

Entradas: La entrada se compone de una cadena de símbolos de longitud mínima 1 y longitud máxima 100, que contiene el código genético de la bacteria. La siguiente entrada es una cadena de largo mínimo 1 y largo máximo 100 que contiene el código genético del virus (ARN). Ambas cadenas están compuestas solo por los caracteres: **A**, **C**, **G** y **T**, lo cual debe ser verificado.

¹ACGT es el acrónimo para los cuatro tipos de bases nitrogenadas que se encuentran en la molécula del ADN: adenina (A), citosina (C), guanina (G) y timina (T)

Salidas: La salida está compuesta por una cadena, correspondiente al ADN bacteriano final y menor de cada bacteria, después de ser atacado por el virus, seguida por el % de preservación (cantidad de componentes de ADN que se conservó después del ataque del virus dividido por la cantidad de componentes de ADN originales multiplicado por 100).

Ejemplo de entrada 1: GCTTTCGACGAT y TCGA

Ejemplo de salida 1: Bacteria: GCTT (33,333%)

Observación: Después de la infección, el virus buscará en todas las partes del ADN de la bacteria que coinciden con su ARN y lo destruirá: GCTT**TCG**ACGAT ==> GCT**TCG**AT ==> GCTT

Ejemplo de entrada 2: GATCGAGCTTCGAA y TCGA

Ejemplo de salida 2: Bacteria: GAGCTA (42,857%)

Ejemplo de entrada 3: GGTCTAGCTAAT y TCGA

Ejemplo de salida 3: Bacteria: GGTCTAGCTAAT (100%)

Requisito: Este problema se debe resolver usando funciones

5.- **Tendencia** Marcar tendencia es uno de los objetivos de muchos usuarios de las redes sociales. En Twitter, las palabras que son tendencia son monitoreadas permanentemente pero hay que ser desconfiado, pues mucho se habla de la manipulación de las redes sociales para conseguir objetivos comerciales, políticos y de otra naturaleza. Jack Dorsey² le ha encargado que diariamente analice los cerca de 65 millones de tuits buscando la palabra que es tendencia. Para ello suponga que:

- los tuits tienen a lo más 280 caracteres
- las palabras vienen separadas solamente por espacios
- interesan sólo palabras de tres o más letras
- se considerará que los tuits vienen sólo en español
- las palabras con mayúsculas y minúsculas que coincidan se considerarán las mismas

Entradas: La entrada al programa corresponde a un conjunto de tuits, el último un mensaje vacío usado para finalizar el ingreso de datos.

Salidas: La única salida del programa es la palabra más citada en los tuits del día.

Ejemplo de entrada 1:

```
Piñera presumiendo sus contratos con Pfizer fin al COVID
Con un siete por ciento Piñera lidera el grupo de presidentes con menor respaldo en la
historia republicana de Chile
Nuevos brotes de covid en Los Ángeles
Tercera ola de Covid en Europa
Vacunas contra el Covid aplicadas en Inglaterra usa y Canadá
Nueva cuarentena en región del Bío Bío por aumento de casos de covid positivo
```

²Creador de Twitter.

Ejemplo de salida 1: covid

- 6.- **Ananagrama** En el dominio de las palabras el concepto de anagrama es bastante conocido, se trata de palabras que se forman con las mismas letras reordenadas, este es el caso de palabras como AMOR / ROMA / MORA / OMAR, NECESARIO / ESCENARIO, PIRATA / PATRIA, CONVERSAR / CONSERVAR, etc. Sin embargo, menos conocido es el concepto de ananagrama, es decir, palabras cuyas letras no producen ninguna otra palabra conocida del lenguaje al ser combinadas de una manera distinta, este sería el caso de OBVIO, SEPULTURA, SIETE, etc. Para ayudar a determinar nuevos vocablos que pudieran ser ananagramas (y anagramas) se le pide que construya un programa Python que, a partir de una palabra P, genere todas las combinaciones posibles de letras de P, estas deben estar enumeradas para que el usuario (experto lingüista) indique cuáles son vocablos válidos en el lenguaje. Así, si el experto señala más de un número (separado por espacios) entonces las palabras P y sus derivadas deberán registrarse como anagramas en cambio si el experto no señala ninguna palabra (ingresa 0) se considerará a P como un ananagrama.

Entradas: La entrada al programa consiste en una palabra P de longitud máxima 10, la cual puede venir en mayúsculas y/o minúsculas. Luego, cuando todas las configuraciones de palabras hayan sido generadas, digamos *n*, el experto ingresará 1 o un conjunto de números separados por espacio indicando las posiciones de las que constituyen anagramas.

Salidas: La salida del programa será una lista de las palabras que son anagramas o sólo la palabra original si es un ananagrama.

Ejemplo de entrada (parte 1): Amor

Ejemplo de salida (parte 1):

- | | |
|----------|----------|
| 1. AMOR | 13. MAOR |
| 2. AMRO | 14. MARO |
| 3. AROM | 15. MORA |
| 4. ARMO | 16. MOAR |
| 5. AORM | 17. MROA |
| 6. AOMR | 18. MRAO |
| 7. RMOA | 19. OMAR |
| 8. RMAO | 20. OMRA |
| 9. RAMO | 21. OARM |
| 10. RAOM | 22. OAMR |
| 11. ROAM | 23. ORAM |
| 12. ROMA | 24. ORMA |

Ejemplo de entrada (parte 2): 1 4 9 12 15 19

Ejemplo de salida (parte 2): Anagramas: AMOR, ARMO, RAMO, ROMA, MORA, OMAR