

1. (8 points) Diseñar y crear una programa que lea un archivo, el cual incluirá en cada fila un identificador de país (expresado por el código del país basado en el estandar ISO 3166-1), seguido por la tasa de crecimiento poblacional (expresado en %). El programa deberá de generar con la información un diccionario, donde la clave sea el código del país y el valor la tasa de crecimiento poblacional y calcular el promedio de la tasa de crecimiento poblacional.

Contenido de archivo Datos.txt :

```
PE  1.29
JP -0.10
AR  1.03
US  0.77
BR  0.92
IN  1.28
CN  0.51
CL  0.91
```

La ejecución sería :

```
Ingrese el nombre del archivo: Datos.txt
La Tasa de Crecimiento Poblacional promedio es: 0.826
```

2. (8 points) Diseñar y crear una programa que lea un archivo, el cual incluirá en cada fila un identificador de país (expresado por el código del país basado en el estandar ISO 3166-1), seguido por la tasa de crecimiento poblacional (expresado en %). El programa deberá de generar con la información un diccionario, donde la clave sea el código del país y el valor la tasa de crecimiento poblacional y presentar el código del país con menor tasa poblacional.

Contenido de archivo Datos.txt :

```
PE  1.29
JP -0.10
AR  1.03
US  0.77
BR  0.92
IN  1.28
CN  0.51
CL  0.91
```

El resultado sería :

```
Ingrese el nombre del archivo: Datos.txt
El pais con menor tasa poblacional es: JP
```

3. (6 points) Diseñe e implemente un programa que pida como dato un número entero **n**, y luego realice lo siguiente:
- Cree un archivo llamado **numeros.txt**, donde se graben **n** números aleatorios un por línea. Los números generados podrán ser desde el 1 al 99.
 - Luego, cree un segundo archivo llamado **filtrados.txt**, en donde se grabe únicamente aquellos números que figuran en el primer archivo y que son **números primos**.

Note, que en la pantalla no se imprimirá nada, ya que la información queda guardada en los archivos: **numeros.txt** y **filtrados.txt**

Recuerde, que esta trabajando con archivos texto, por lo cual **todo lo que se graba o lee son cadenas**.

Algunos ejemplos de diálogo de este programa serían:

Listing 1: Ejemplo 1

```
Input:
Numero de elementos : 12
Output:
Listo los dos archivos creados!!!

En su dispositivo de almacenamiento aparecera:

numeros.tx
87
50
38
7
47
48
17
54
72
39
51
18

filtados.txt
7
47
```

17

Listing 2: Ejemplo 2

```
Input:
Numero de elementos : 20

Output:
Listo los dos archivos creados!!!

En su dispositivo de almacenamiento aparecera:

numeros.txt

43
49
66
88
46
57
60
70
70
17
47
69
56
49
62
37
29
41
68
94

filtrados.txt
43
17
47
37
29
41
```

4. Haciendo uso de un diccionario, implemente un algoritmo que permita determinar cuántas veces se repite cada carácter de un string. Considere que su algoritmo debe considerar

cualquier carácter (letras, números, símbolos, etc.) excepto espacios en blanco.

Algunos ejemplos de diálogo de este programa serían:

```
Input: aaabbaa
Output: a: 5, b: 2
```

```
Input: hola como estas
Output: h: 1, o: 3, l: 1, a: 2, c: 1, m: 1, e:1, s: 2
```

```
Ingrese frase: Sacare 20 en esta practica
Output: {"s": 2, "a": 5, "c": 3, "r": 2, "e": 3, "2": 1,
        "0": 1, "n": 1, "t": 2, "i": 1}
```

5. (Nivel 1) Implemente un programa en Python que contenga una función que reciba una lista de números decimales y muestre en pantalla la cantidad de veces que se repite cada número de la lista.

Por ejemplo:

```
Ingrese la lista de decimales: [1.2, 1.3, 1.2, 0.5]
1.2 = 2
1.3 = 1
0.5 = 1
```

```
Ingrese la lista de decimales: [1.2, 1.2, 1.2, 1.2, 1.2]
1.2 = 5
```

6. (Nivel 1) Use un diccionario para crear un traductor de números del 1 al 10 al japonés. Tomar en cuenta la siguiente equivalencia.

```
Ichii
Ni
San
Yon
Go
Roku
Nana
Hachi
Kyu
Ju
```

Veamos algunos ejemplos:

```
input:
3
output:
San
```

```
input:
5
output:
Go
```

7. (Nivel 2) Diseñe e implemente un algoritmo que reciba como dato de entrada un texto, y proceda a extraer todas las palabras del texto asociando la cantidad de veces que se repite dicha palabra en el texto. Use un **diccionario** para guardar cada palabra diferente y las veces que se repite.

Algunos ejemplos de diálogo de este programa serían:

```
Input: Hola Como Estas hola como estas HOLA
Output: {'hola': 3, 'como': 2, 'estas': 2}
```

```
Input: Cuando cuentas cuantos cuenta cuantos cuantos cuentas
       porque si cuentas cuantos cuantos cuentas sabras cuantos
       cuantos sabes contar
Output: {'cuando': 1, 'cuentas': 1, 'cuantos': 4, 'cuenta': 1,
        'cuantos': 3, 'cuentas': 3, 'porque': 1, 'si': 1, 'sabras': 1,
        'sabes': 1, 'contar': 1}
```

8. (Nivel 2) Dada una *cadena* de ADN generar su *diccionario* que tenga el primer y último elemento de la cadena e indique cuántas veces aparece la primera y última base de la cadena.

Veamos algunos ejemplos:

```
input:
cccaactgaa
output:
{'c': 4, 'a': 4}
```

```
input:
gtactcgggt
output:
{'g': 4, 't': 3}
```

9. (Nivel 3) Tienes un diccionario donde cada key es el nombre de un alumno y cada valor es una lista con las comidas que le gustan. Crea un programa que te diga qué alumnos pueden comer juntos. Solo pueden comer juntos aquellos que compartan una comida en común.

Por ejemplo, si el diccionario es el siguiente:

```
alumnos = { "Alejandro": ["ceviche", "lomo saltado", "pollo a la brasa"],
            "Roberto": ["lomo saltado"],
            "Jes s": ["lomo saltado", "lasagna"],
            "Carlos": ["lasagna"],
            "Melina": ["ceviche", "arroz con pollo"]}
```

Un ejemplo de diálogo de este programa sería:

```
Pueden comer juntos:
Alejandro, Roberto, Jes s
Jes s y Carlos
Melina y Alejandro
```

10. (Nivel 4) **Usando listas por comprensión**, genera una lista con 10 valores al azar entre 1 y 100. Luego, también usando comprensión, crea una lista con los cuadrados de la lista anterior. Finalmente, crea una lista que represente una matriz de 4 x 4 cuyos valores sean los valores de las dos listas anteriores, pero que no tenga los 4 valores más pequeños.

Por ejemplo:

```
Lista de 10 valores al azar:
[25, 48, 0, 84, 52, 51, 10, 46, 43, 2]
Lista de cuadrados:
[625, 2304, 0, 7056, 2704, 2601, 100, 2116, 1849, 4]

Matriz 4x4:
  25   48   84   52
  51   10   46   43
 625 2304 7056 2704
2601  100 2116 1849
```