Ejercicio 6. (3 puntos)

Hacer un programa que solicite al usuario el ingreso de dos números enteros. El programa debe crear un archivo con nombre "numeros.txt" y escribir todos los números en el intervalo cerrado entre los dos valores ingresados.

El programa debe validar en el ingreso que:
* Ambos números sean enteros
* Los números son distintos entre sí. Si son iguales debe pedir solamente el segundo número.

Si el primer número es mayor que el segundo, los números deben escribirse en forma descendente. En caso contrario, los números deben escribirse en forma ascendente.

El formato del archivo debe ser:
* Los números deben escribirse separados por un espacio
* Deben escribirse 10 números por línea, excepto la última línea, que contendrá los restantes.

Calificación:
* Valida correctamente los ingresos y los solicita nuevamente si hay un error o son iguales según consigna (1 punto)
* Escribe la secuencia correcta de números en el órden correcto (1 punto)
* El archivo cumple con el formato solicitado (1 punto).

Recuerde que su programa debe ser funcional.

Ejemplo:

```
Ingrese un número: a34
Ingreso inválido!!
Ingrese un número: 34
Ingrese otro número distinto al primero: 34
Ingreso inválido!!
Ingrese otro número distinto al primero: 5

Se generó el archivo!!
```

Y el archivo generado contiene lo siguiente:

```
34 33 32 31 30 29 28 27 26 25
24 23 22 21 20 19 18 17 16 15
14 13 12 11 10 9 8 7 6 5
```

Ejemplo:
```
Ingrese un número: 90
Ingrese otro número distinto al primero: 112
```

```
Se generó el archivo!!
```
Y el archivo generado contiene lo siguiente:

```
90 91 92 93 94 95 96 97 98 99
100 101 102 103 104 105 106 107 108 109
110 111 112
```

Ejercicio 7. (3 puntos)

Dada una lista compuesta únicamente de palabras (cadenas) que pueden repetirse, contá el número de ocurrencias de cada una de ellas.
Imprime en pantalla:

- La cantidad de palabras únicas
- el número de ocurrencias **de mayor a menor** de cada palabra. En caso de empate de ocurrencias desempata el orden de aparición de la palabra en la lista original.

Restricciones

- **Debes dividir el problema en partes usando no menos de dos funciones.**
- La lista original no debe alterarse.
- La respuesta debe funcionar para cualquier cantidad de palabras y para la lista vacía.
- El ejemplo es a fines ilustrativos. Debe funcionar con cualquier lista de palabras.

Calificación:
* Generar correctamente las palabras únicas (1 punto)
* Generar correctamente la cantidad de palabras únicas (1 punto).
* Escribir ordenado de mayor a menor el número de ocurrencias de cada palabra, segun especificaciones dadas(1 punto)

Ejemplo

Entrada

```
palabras = [“naranja”,”mandarina”,”mandarina”,”frutilla”,”melon”
             ,”melon”,”melon”,”kiwi”,”kiwi”]
```

Salida

```
5
melon 3
mandarina 2
kiwi 2
naranja 1
```

frutilla 1

Ejercicio 8. (2 puntos)

Hacer una función que dada una frase con palabras, calcule la cantidad total de veces que aparece en toda la frase cada una de las vocales y cual es la vocal que más se repite.

Debe tener presente que la letra ‘A’ es la misma vocal que la ‘a’ y así con todas las vocales.

No hay que hacer todo el programa.
Una función puede llamar a otras.

Calificación:
* Calcular la cantidad de veces que se repiten las vocales en la frase dada (1 punto)
* Calcular la letra que más se repite (1 punto).

Ejemplo:

```
frase = 'Hoy puede ser un Gran Dia planteatelo asi...
Aprovecharlo o que pase de largo, depende en parte de ti'
```

data=mostrara(frase)

La a se repite 10 veces
La e se repite 15 veces
La i se repite 3 veces
La o se repite 6 veces
La u se repite 3 veces

La función mostrará:

```
La vocal que más se repite es la e
```