

## 2.- Apartado 2: Resolver ciertos problemas en Python.

Dado que a lo largo del año vamos a tener que trabajar bastante con Python, es necesario tener cierta base sobre los aspectos básicos del lenguaje. Para ello se propone la realización de los siguientes ejercicios que deberán ser subidos al repositorio GitHub del Apartado 1.

- **Problema 1. División de una lista de enteros.**

Escribe una función que reciba por parámetro una lista de enteros y devuelva dos listas: una con los valores negativos que tuviera y otra con los positivos. Ambas listas deben estar ordenadas ascendentemente.

- **Problema 2. Frecuencia de palabras en un texto.**

Escribe un programa que pida al usuario ingresar una frase o párrafo. Luego, el programa debe contar cuántas veces aparece cada palabra en el texto y mostrar las palabras junto con su frecuencia.

Requisitos:

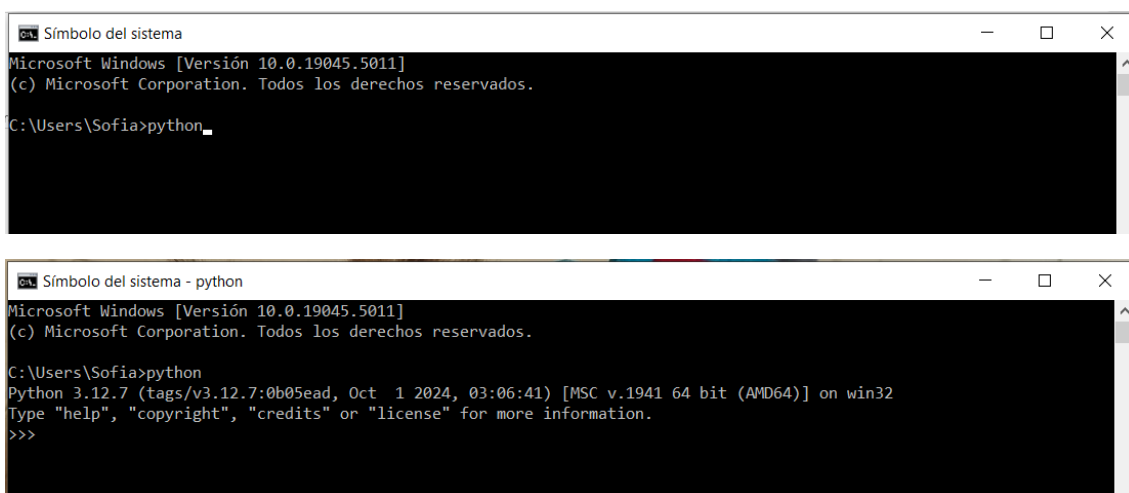
1. Eliminar los signos de puntuación y convertir todas las palabras a minúsculas para evitar diferencias.
2. Usar un diccionario donde la clave sea la palabra y el valor sea su frecuencia.
3. Mostrar las palabras y sus frecuencias de forma ordenada por la palabra.

- **Problema 3. Intersección y unión de conjuntos**

Escribe un programa que permita al usuario crear dos conjuntos de números enteros. Luego, el programa debe calcular y mostrar:

1. La **intersección** de ambos conjuntos (elementos comunes).
2. La **unión** de ambos conjuntos (todos los elementos sin duplicados).
3. La **diferencia simétrica** (elementos que están en uno u otro conjunto, pero no en ambos).

Abrimos Python:



```
Símbolo del sistema
Microsoft Windows [Versión 10.0.19045.5011]
(c) Microsoft Corporation. Todos los derechos reservados.
C:\Users\Sofia>python

Símbolo del sistema - python
Microsoft Windows [Versión 10.0.19045.5011]
(c) Microsoft Corporation. Todos los derechos reservados.
C:\Users\Sofia>python
Python 3.12.7 (tags/v3.12.7:0b05ead, Oct 1 2024, 03:06:41) [MSC v.1941 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

## 2.1.- Problema 1. División de una lista de enteros.

Escribe una función que reciba por parámetro una lista de enteros y devuelva dos listas: una con los valores negativos que tuviera y otra con los positivos. Ambas listas deben estar ordenadas ascendentemente.

### CÓDIGO

```
# LISTA INICIAL CON UNOS NÚMEROS Y DOS VACÍAS EN LAS QUE SE GUARDARN LOS POSITIVOS Y LOS NEGATIVOS
numeros = [0, 58, -94, 21, -66, -3, 0, 7, 33, 42, -3, -11, -87, 66, 23, 96]
positivos = []
negativos = []

# FUNCIÓN PARA SEPARA LOS ELEMENTOS NUMÉRICOS DE LA LISTA (el 0 no es positivo ni negativo)
def separarNumeros(lista):
    for elemento in lista:
        if elemento == 0:
            print("Se ha encontrado el 0, no es ni positivo ni negativo")
            print("-----")
        elif elemento > 0:
            positivos.append(elemento)
        else:
            negativos.append(elemento)

#MOSTRAMOS LA LISTA INICIAL
print("Lista completa de números: ")
print(numeros)
print("-----")

#LLAMAMOS A LA FUNCIÓN PASÁNDOLE LA LISTA COMO PARÁMETRO
separarNumeros(numeros)

#MOSTRAMOS LAS DOS LISTAS FINALES
print("Lista de números positivos: ")
print(positivos)
print("-----")
print("Lista de números negativos: ")
print(negativos)
print("-----")
```

El código como imagen desde el VSCode:

```
problema01.py X
problema01.py > ...
1  # LISTA INICIAL CON UNOS NÚMEROS Y DOS VACÍAS EN LAS QUE SE GUARDARN LOS POSITIVOS Y LOS NEGATIVOS
2  numeros = [0, 58, -94, 21, -66, -3, 0, 7, 33, 42, -3, -11, -87, 66, 23, 96]
3  positivos = []
4  negativos = []
5
6  # FUNCIÓN PARA SEPARA LOS ELEMENTOS NUMÉRICOS DE LA LISTA (el 0 no es positivo ni negativo)
7  def separarNumeros(lista):
8      for elemento in lista:
9          if elemento == 0:
10             print("Se ha encontrado el 0, no es ni positivo ni negativo")
11             print("-----")
12             elif elemento > 0:
13                 positivos.append(elemento)
14             else:
15                 negativos.append(elemento)
16
17  #MOSTRAMOS LA LISTA INICIAL
18  print("Lista completa de números: ")
19  print(numeros)
20  print("-----")
21
22  #LLAMAMOS A LA FUNCIÓN PASÁNDOLE LA LISTA COMO PARÁMETRO
23  separarNumeros(numeros)
24
25  #MOSTRAMOS LAS DOS LISTAS FINALES
26  print("Lista de números positivos: ")
27  print(positivos)
28  print("-----")
29  print("Lista de números negativos: ")
30  print(negativos)
31  print("-----")
32
```

Salida desde terminal de VSCode:

```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS

PS C:\Users\Sofia\Documents\python2024\pia> python problema01.py
Lista completa de números:
[0, 58, -94, 21, -66, -3, 0, 7, 33, 42, -3, -11, -87, 66, 23, 96]
-----
Se ha encontrado el 0, no es ni positivo ni negativo
-----
Se ha encontrado el 0, no es ni positivo ni negativo
-----
Lista de números positivos:
[58, 21, 7, 33, 42, 66, 23, 96]
-----
Lista de números negativos:
[-94, -66, -3, -3, -11, -87]
-----
PS C:\Users\Sofia\Documents\python2024\pia>
```

Salida vista desde CMD:

```
ca. Símbolo del sistema
Microsoft Windows [Versión 10.0.19045.5011]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Sofia>cd documents\python2024\pia

C:\Users\Sofia\Documents\python2024\pia>python problema01.py
Lista completa de números:
[0, 58, -94, 21, -66, -3, 0, 7, 33, 42, -3, -11, -87, 66, 23, 96]
-----
Se ha encontrado el 0, no es ni positivo ni negativo
-----
Se ha encontrado el 0, no es ni positivo ni negativo
-----
Lista de números positivos:
[58, 21, 7, 33, 42, 66, 23, 96]
-----
Lista de números negativos:
[-94, -66, -3, -3, -11, -87]
-----

C:\Users\Sofia\Documents\python2024\pia>
```

## 2.2.- Problema 2. Frecuencia de palabras en un texto.

Escribe un programa que pida al usuario ingresar una frase o párrafo. Luego, el programa debe contar cuántas veces aparece cada palabra en el texto y mostrar las palabras junto con su frecuencia.

Requisitos:

1. Eliminar los signos de puntuación y convertir todas las palabras a minúsculas para evitar diferencias.
2. Usar un diccionario donde la clave sea la palabra y el valor sea su frecuencia.
3. Mostrar las palabras y sus frecuencias de forma ordenada por la palabra.

### CÓDIGO

```
#LIBRERIAS NECESARIAS
import re

#PETICIÓN DE FRASE AL USUARIO
frase = input("Escribe una frase: ")
print("-----")
print("Frase introducida: ", frase)
print("-----")

#ELIMINACIÓN DE SIGNOS DE PUNTUACIÓN
texto = re.sub(r'[^\w\s]', '', frase)
print("Frase sin signos de puntuación: ", texto)
print("-----")

#CONVERSIÓN A MINÚSCULAS
textoMinus = texto.lower()
print("Frase sin signos de puntuación y en minúsculas: ", textoMinus)
print("-----")

#SEPARACIÓN DEL TEXTO EN PALABRAS EN UNA LISTA
lista = textoMinus.split()
print("Lista: ", lista)
print("-----")

#ORDENACIÓN DE LA LISTA DE PALABRAS
listaOrdenada = lista.sort()
print("Lista ordenada: ", listaOrdenada)
print("-----")

#CÁLCULO DE LA FRECUENCIA DE CADA PALABRA
frecuenciaPalab = []
for frec in lista:
    frecuenciaPalab.append(lista.count(frec))
print("Frecuencia palabras: ", frecuenciaPalab)
print("-----")

#CREAMOS UN DICCIONARIO UNIENDO DOS LISTAS DE LA MISMA LONGITUD (ya no tiene duplicados)
diccionario = zip(lista, frecuenciaPalab)
dicc01 = dict(diccionario) #Para mostrarlo bien se ha de convertir de zip a diccionario
print("Diccionario: ", dicc01)
print("-----")
```

El código como imagen desde el VSCode:

```
problema02.py X
problema02.py > ...
1  #LIBRERIAS NECESARIAS
2  import re
3  #PETICIÓN DE FRASE AL USUARIO
4  frase = input("Escribe una frase: ")
5  print("-----")
6  print("Frase introducida: ", frase)
7  print("-----")
8  #ELIMINACIÓN DE SIGNOS DE PUNTUACIÓN
9  texto = re.sub(r'[^\w\s]', '', frase)
10 print("Frase sin signos de puntuación: ", texto)
11 print("-----")
12 #CONVERSIÓN A MINÚSCULAS
13 textoMinus = texto.lower()
14 print("Frase sin signos de puntuación y en minúsculas: ", textoMinus)
15 print("-----")
16 #SEPARACIÓN DEL TEXTO EN PALABRAS EN UNA LISTA
17 lista = textoMinus.split()
18 print("Lista: ", lista)
19 print("-----")
20 #ORDENACIÓN DE LA LISTA DE PALABRAS
21 listaOrdenada = lista.sort()
22 print("Lista ordenada: ", listaOrdenada)
23 print("-----")
24 #CÁLCULO DE LA FRECUENCIA DE CADA PALABRA
25 frecuenciaPalab = []
26 for frec in lista:
27     frecuenciaPalab.append(lista.count(frec))
28 print("Frecuencia palabras: ", frecuenciaPalab)
29 print("-----")
30 #CREAMOS UN DICCIONARIO UNIENDO DOS LISTAS DE LA MISMA LONGITUD (ya no tiene duplicados)
31 diccionario = zip(lista, frecuenciaPalab)
32 dicc01 = dict(diccionario) #Para mostrarlo bien se ha de convertir de zip a diccionario
33 print("Diccionario: ", dicc01)
34 print("-----")
35
```

Salida desde terminal de VSCode:

```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS
PS C:\Users\Sofia\Documents\python2024\pia> python problema02.py
Escribe una frase: La casa de mi ABUEla estaba, en -mi linda Cordobita! Que linda es mi ciudad.
Frase introducida: La casa de mi ABUEla estaba, en -mi linda Cordobita! Que linda es mi ciudad.
Frase sin signos de puntuación: La casa de mi ABUEla estaba en mi linda Cordobita Que linda es mi ciudad
Frase sin signos de puntuación y en minúsculas: la casa de mi abuela estaba en mi linda cordobita que linda es mi ciudad
Lista: ['la', 'casa', 'de', 'mi', 'abuela', 'estaba', 'en', 'mi', 'linda', 'cordobita', 'que', 'linda', 'es', 'mi', 'ciudad']
Lista ordenada: None
Frecuencia palabras: [1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 3, 3, 3, 1]
Diccionario: {'abuela': 1, 'casa': 1, 'ciudad': 1, 'cordobita': 1, 'de': 1, 'en': 1, 'es': 1, 'estaba': 1, 'la': 1, 'linda': 2, 'mi': 3, 'que': 1}
PS C:\Users\Sofia\Documents\python2024\pia>
```

Salida vista desde CMD:

```
Microsoft Windows [Versión 10.0.19045.5011]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Sofia>cd documents\python2024\pia

C:\Users\Sofia\Documents\python2024\pia>python problema02.py
Escribe una frase: La casa de mi ABUEla estaba, en -mi linda Cordobita! Que linda es mi ciudad.
Frase introducida: La casa de mi ABUEla estaba, en -mi linda Cordobita! Que linda es mi ciudad.
Frase sin signos de puntuación: La casa de mi ABUEla estaba en mi linda Cordobita Que linda es mi ciudad
Frase sin signos de puntuación y en minúsculas: la casa de mi abuela estaba en mi linda cordobita que linda es mi ciudad
Lista: ['la', 'casa', 'de', 'mi', 'abuela', 'estaba', 'en', 'mi', 'linda', 'cordobita', 'que', 'linda', 'es', 'mi', 'ciudad']
Lista ordenada: None
Frecuencia palabras: [1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 3, 3, 3, 1]
Diccionario: {'abuela': 1, 'casa': 1, 'ciudad': 1, 'cordobita': 1, 'de': 1, 'en': 1, 'es': 1, 'estaba': 1, 'la': 1, 'linda': 2, 'mi': 3, 'que': 1}

C:\Users\Sofia\Documents\python2024\pia>
```

### 2.3.- Problema 3. Intersección y unión de conjuntos.

Escribe un programa que permita al usuario crear dos conjuntos de números enteros. Luego, el programa debe calcular y mostrar:

1. La **intersección** de ambos conjuntos (elementos comunes).
2. La **unión** de ambos conjuntos (todos los elementos sin duplicados).
3. La **diferencia simétrica** (elementos que están en uno u otro conjunto, pero no en ambos).

#### CÓDIGO

```
#IMPORTACIONES DE MÓDULOS NECESARIOS
import re

#PATRÓN PARA CONTROL DE ENTRADA
patron = re.compile("^[xX]$| ^[-?\d+$]")

#CONJUNTOS DE DATOS UTILIZADOS
conjuntoAux = set()
conjunto1 = set()
conjunto2 = set()
interseccion = set()
union = set()
diferenciaSimetrica = set()

#FUNCIÓN DE COMPROBACIÓN DE LOS DATOS INTRODUCIDOS Y AÑADIDOS A LOS CONJUNTOS
def comprobar():
    while True:
        entrada = input()
        if re.match(patron, entrada):
            if (entrada == "x" or entrada == "X"):
                print("Se cierra el conjunto")
                print("Conjunto auxiliar cerrado:\t", conjuntoAux)
                print("-----")
                break
            else:
                conjuntoAux.add(entrada)
        else:
            print("*** EL VALOR INTRODUCIDO NO ES VÁLIDO ***")
            print("*** Por favor introduzca un número entero o 'x' para cerrar el conjunto ***")
            continue

#PETICIÓN DE NÚMEROS PARA EL CONJUNTO 1
print("Introduce los números enteros para formar el conjunto 1 ('x' para finalizar)")
comprobar()
conjunto1 = conjuntoAux.copy()
conjuntoAux.clear()

#PETICIÓN DE NÚMEROS PARA EL CONJUNTO 2
print("Introduce los números enteros para formar el conjunto 2 ('x' para finalizar)")
comprobar()
conjunto2 = conjuntoAux.copy()
conjuntoAux.clear()

#SALIDA DE AMBOS CONJUNTOS POR PANTALLA
print("Conjunto 1:\t", conjunto1)
```

```
print("-----")
print("Conjunto 2:\t", conjunto2)
print("-----")

#INTERSECCIÓN DE AMBOS CONJUNTOS
interseccion = conjunto1.intersection(conjunto2)
print("Intersección de ambos conjuntos:\t", interseccion)
print("-----")

#UNIÓN DE AMBOS CONJUNTOS
union = conjunto1.union(conjunto2)
print("Unión de ambos conjuntos:\t", union)
print("-----")

#DIFERENCIA SIMÉTRICA DE AMBOS CONJUNTOS
diferenciaSimetrica = conjunto1.symmetric_difference(conjunto2)
print("Diferencia simétrica de ambos conjuntos:\t", diferenciaSimetrica)
print("-----")
```

El código como imagen desde el VSCode:



```
problema03.py X
problema03.py > ...
1  #IMPORTACIONES DE MÓDULOS NECESARIOS
2  import re
3
4  #PATRÓN PARA CONTROL DE ENTRADA
5  patron = re.compile("[xx]^[^-\d+]+$")
6
7  #CONJUNTOS DE DATOS UTILIZADOS
8  conjuntoAux = set()
9  conjunto1 = set()
10 conjunto2 = set()
11 interseccion = set()
12 union = set()
13 diferenciaSimetrica = set()
14
15 #FUNCIÓN DE COMPROBACIÓN DE LOS DATOS INTRODUCIDOS Y AÑADIDOS A LOS CONJUNTOS
16 def comprobar():
17     while True:
18         entrada = input()
19         if re.match(patron, entrada):
20             if (entrada == "x" or entrada == "X"):
21                 print("Se cierra el conjunto")
22                 print("Conjunto auxiliar cerrado:\t", conjuntoAux)
23                 print("-----")
24                 break
25             else:
26                 conjuntoAux.add(entrada)
27         else:
28             print("*** EL VALOR INTRODUCIDO NO ES VÁLIDO ***")
29             print("*** Por favor introduzca un número entero o 'x' para cerrar el conjunto ***")
30             continue
31
32 #PETICIÓN DE NÚMEROS PARA EL CONJUNTO 1
33 print("Introduce los números enteros para formar el conjunto 1 ('x' para finalizar)")
34 comprobar()
35 conjunto1 = conjuntoAux.copy()
36 conjuntoAux.clear()
37
38 #PETICIÓN DE NÚMEROS PARA EL CONJUNTO 2
39 print("Introduce los números enteros para formar el conjunto 2 ('x' para finalizar)")
40 comprobar()
41 conjunto2 = conjuntoAux.copy()
42 conjuntoAux.clear()
43
44 #SALIDA DE AMBOS CONJUNTOS POR PANTALLA
45 print("Conjunto 1:\t", conjunto1)
46 print("-----")
47 print("Conjunto 2:\t", conjunto2)
48 print("-----")
49
50 #INTERSECCIÓN DE AMBOS CONJUNTOS
51 interseccion = conjunto1.intersection(conjun (variable) interseccion: set
52 print("Intersección de ambos conjuntos:\t", interseccion)
53 print("-----")
54
55 #UNIÓN DE AMBOS CONJUNTOS
56 union = conjunto1.union(conjunto2)
57 print("Unión de ambos conjuntos:\t", union)
58 print("-----")
59
60 #DIFERENCIA SIMÉTRICA DE AMBOS CONJUNTOS
61 diferenciaSimetrica = conjunto1.symmetric_difference(conjunto2)
62 print("Diferencia simétrica de ambos conjuntos:\t", diferenciaSimetrica)
63 print("-----")
64
```

## Salida desde terminal de VSCode:

```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS

PS C:\Users\Sofia\Documents\python2024\pia> python problema03.py
C:\Users\Sofia\Documents\python2024\pia\problema03.py:5: SyntaxWarning: invalid escape sequence '\d'
  patron = re.compile("^[\d$]{1-?}\d+$")
Introduce los números enteros para formar el conjunto 1 ('x' para finalizar)
14
-9
55
** EL VALOR INTRODUCIDO NO ES VÁLIDO **
** Por favor introduzca un número entero o 'x' para cerrar el conjunto **
-22
20
-3
80
99
x
Se cierra el conjunto
Conjunto auxiliar cerrado:      {'-3', '14', '80', '-9', '99', '20', '-22'}
-----
Introduce los números enteros para formar el conjunto 2 ('x' para finalizar)
-9
44
-33
99
11
-47
-22
-80
x
Se cierra el conjunto
Conjunto auxiliar cerrado:      {'44', '11', '-47', '-9', '99', '-80', '-22', '-33'}
-----
Conjunto 1:      {'-3', '14', '-9', '80', '99', '20', '-22'}
-----
Conjunto 2:      {'44', '11', '-47', '-9', '99', '-80', '-22', '-33'}
-----
Intersección de ambos conjuntos:      {'-9', '-22', '99'}
-----
Unión de ambos conjuntos:      {'44', '-3', '14', '80', '11', '-47', '-9', '99', '20', '-80', '-22', '-33'}
-----
Diferencia simétrica de ambos conjuntos:      {'44', '-3', '14', '11', '80', '-47', '20', '-80', '-33'}
-----
PS C:\Users\Sofia\Documents\python2024\pia>
```

## Salida vista desde CMD:

```
Simbolo del sistema
C:\Users\Sofia>cd documents/python2024/pia

C:\Users\Sofia\Documents\python2024\pia>python problema03.py
C:\Users\Sofia\Documents\python2024\pia\problema03.py:5: SyntaxWarning: invalid escape sequence '\d'
  patron = re.compile("^[\d$]{1-?}\d+$")
Introduce los números enteros para formar el conjunto 1 ('x' para finalizar)
11
99
-3
-77
69
20
-45
91
-33
x
Se cierra el conjunto
Conjunto auxiliar cerrado:      {'69', '-77', '-33', '11', '-45', '-20', '99', '-3', '91'}
-----
Introduce los números enteros para formar el conjunto 2 ('x' para finalizar)
9
-20
1
** EL VALOR INTRODUCIDO NO ES VÁLIDO **
** Por favor introduzca un número entero o 'x' para cerrar el conjunto **
-33
91
13
77
-5
50
x
Se cierra el conjunto
Conjunto auxiliar cerrado:      {'13', '77', '50', '9', '-20', '-5', '-33', '91'}
-----
Conjunto 1:      {'69', '-77', '-33', '11', '-45', '-20', '99', '-3', '91'}
-----
Conjunto 2:      {'13', '77', '50', '9', '-20', '-5', '-33', '91'}
-----
Intersección de ambos conjuntos:      {'-33', '-20', '91'}
-----
Unión de ambos conjuntos:      {'69', '-77', '-33', '13', '77', '50', '11', '9', '-45', '-20', '99', '-5', '-3', '91'}
-----
Diferencia simétrica de ambos conjuntos:      {'69', '-77', '13', '77', '50', '11', '9', '-45', '99', '-5', '-3'}
-----
C:\Users\Sofia\Documents\python2024\pia>
```