

Prueba del problema 1:

```
Problema1.py > ...
1  numeros_enteros=[4, -1, 2, 4, 3, -5, 2 ]
2
3  def procesar_numeros_enteros(numeros_enteros):
4      for numero in numeros_enteros:
5          if numero < 0:
6              numeros_enteros.remove(numero)
7      sin_repetir_y_ordenados=sorted(list(set(numeros_enteros)))
8      return sin_repetir_y_ordenados
9
10 print(procesar_numeros_enteros(numeros_enteros))
11
12
13
14
15
16
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\Memory\Documents\VisualStudioCodeProjects\Tarea 1 Pia> & C:
13/python.exe "c:/Users/Memory/Documents/VisualStudioCodeProjects/Tarea
[2, 3, 4]
PS C:\Users\Memory\Documents\VisualStudioCodeProjects\Tarea 1 Pia>
```

Prueba del problema 2:

```
Problema2.py > ...
1  import string
2
3  lista_palabras = ["caso", "práctico", "elena", "de", "almacén", "clientes"]
4  ruta_archivo = "Prueba.txt"
5
6  def contar_frecuencia_palabras(lista_palabras, ruta_archivo):
7      # Inicializa un diccionario para almacenar la frecuencia
8      frecuencias = {palabra: 0 for palabra in lista_palabras}
9
10     # Lee el contenido del archivo
11     with open(ruta_archivo, 'r', encoding='utf-8') as archivo:
12         contenido = archivo.read()
13
14     # Elimina signos de puntuación y convierte a minúsculas
15     contenido = contenido.translate(str.maketrans('', '', string.punctuation)).lower()
16
17     # Cuenta la frecuencia de cada palabra en la lista
18     for palabra in lista_palabras:
19         frecuencia = contenido.count(palabra)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python

```
documents\VisualStudioCodeProjects\Tarea 1 Pia> & C:/Users/Memory/AppData/Local/Programs/Python/Python313/python
ers/Memory/Documents/VisualStudioCodeProjects/Tarea 1 Pia/Problema2.py"
almacén: 1
caso: 1
clientes: 4
de: 42
elena: 7
práctico: 1
PS C:\Users\Memory\Documents\VisualStudioCodeProjects\Tarea 1 Pia>
```

Prueba del problema 3:

```
Problema3.py > operaciones_con_juntos
1  def operaciones_con_juntos(lista1, lista2):
2      # Convierte las listas en conjuntos
3      conjunto1 = set(lista1)
4      conjunto2 = set(lista2)
5
6      # Calcula la intersección usando conjuntos
7      interseccion = conjunto1.intersection(conjunto2)
8
9      # Calcula la unión usando conjuntos
10     union = conjunto1.union(conjunto2)
11
12     # Calcula la diferencia simétrica usando conjuntos
13     diferencia_simetrica = conjunto1.symmetric_difference(conjunto2)
14
15     # Crea el diccionario con los resultados
16     resultados = {
17         "interseccion": list(interseccion),
18         "union": list(union),
19         "diferencia simetrica": list(diferencia_simetrica)
20     }
21     return resultados
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
ocuments\VisualStudioCodeProjects\Tarea 1 Pia> & C:/Users/Memory/AppData/Local/Pr
ers/Memory/Documents/VisualStudioCodeProjects/Tarea 1 Pia/Problema3.py"
Intersección: [4, 5]
Unión: [1, 2, 3, 4, 5, 6, 7, 8]
Diferencia Simétrica: [1, 2, 3, 6, 7, 8]
PS C:\Users\Memory\Documents\VisualStudioCodeProjects\Tarea 1 Pia>
```