Tecnológico de Monterrey

# Actividad 4.
# Documentación de Ejercicios

A01794935 - Sergio Enrique Pulido Morales

El presente trabajo ha sido realizado cumpliendo las políticas del curso y con los criterios de evaluación de la actividad. Asimismo, establezco que el contenido de este trabajo ha sido documentado en fuentes bibliográficas autorizadas, por tanto, la información redactada no ha sido plagiada de otro documento o trabajo ajeno ni de cualquier otra fuente de carácter confidencial.

**Pruebas de Software y Aseguramiento de la Calidad** | Fecha: 30/01/2025

## Exercise #1: Compute statistics

☒Req1. The program shall be invoked from a command line. The program shall receive a file as parameter. The file will contain a list of items (presumable numbers).

☒Req 2. The program shall compute all descriptive statistics from a file containing numbers. The results shall be print on a screen and on a file named StatisticsResults.txt. All computation MUST be calculated using the basic algorithms, not functions or libraries. The descriptive statistics are mean, median, mode, standard deviation, and variance.

☒Req 3. The program shall include the mechanism to handle invalid data in the file. Errors should be displayed in the console and the execution must continue.

☒Req 4. The name of the program shall be computeStatistics.py

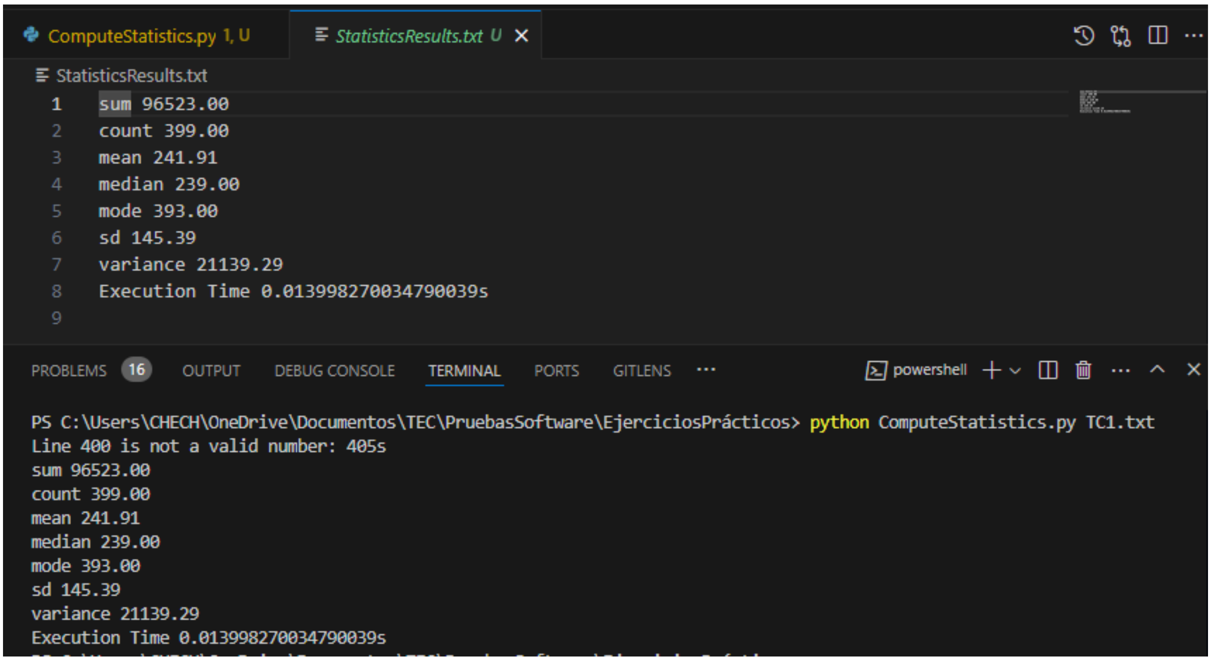
☒Req 5. The minimum format to invoke the program shall be as follows: python computeStatistics.py fileWithData.txt

☒Req 6. The program shall manage files having from hundreds of items to thousands of items.

☒Req 7. The program should include at the end of the execution the time elapsed for the execution and calculus of the data. This number shall be included in the results file and on the screen.

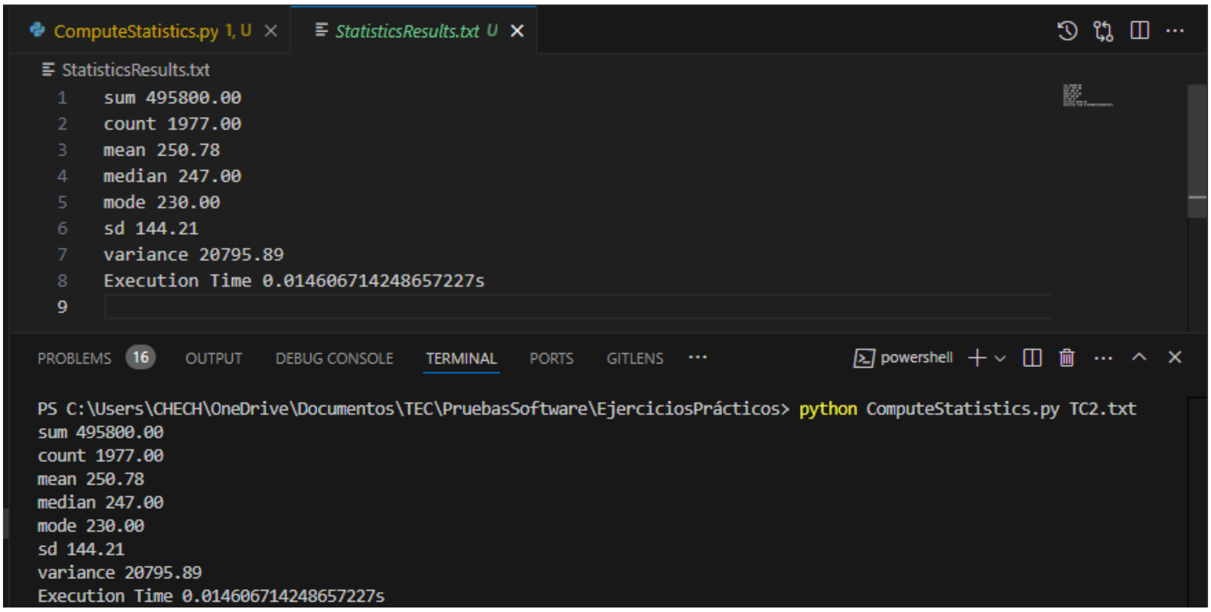
☒Req 8. Be compliant with PEP8

- ### Results



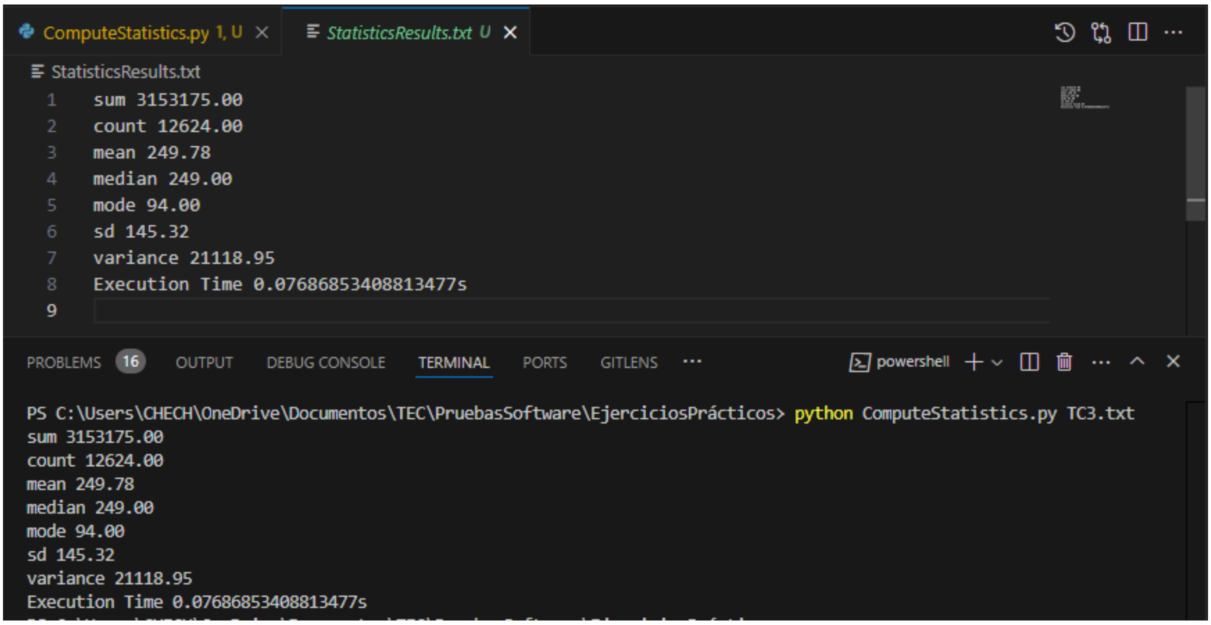
```
sum 96523.00
count 399.00
mean 241.91
median 239.00
mode 393.00
sd 145.39
variance 21139.29
Execution Time 0.013998270034790039s
```

```
PS C:\Users\CHECH\OneDrive\Documentos\TEC\PruebasSoftware\EjerciciosPrácticos> python ComputeStatistics.py TC1.txt
Line 400 is not a valid number: 405s
sum 96523.00
count 399.00
mean 241.91
median 239.00
mode 393.00
sd 145.39
variance 21139.29
Execution Time 0.013998270034790039s
```



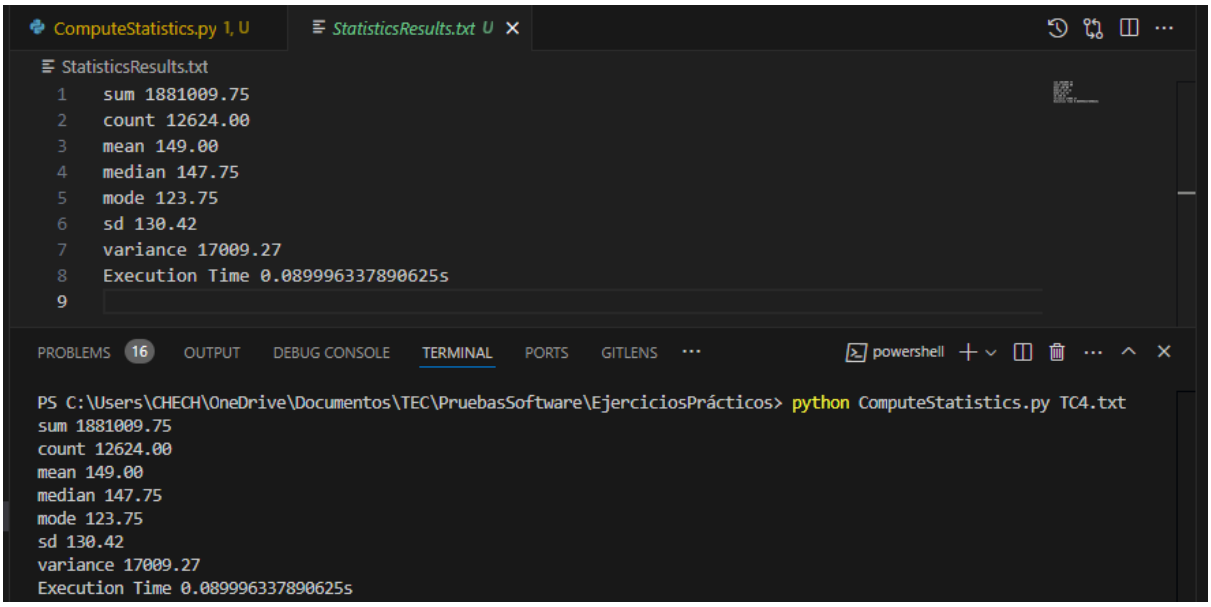
```
sum 495800.00
count 1977.00
mean 250.78
median 247.00
mode 230.00
sd 144.21
variance 20795.89
Execution Time 0.014606714248657227s
```

```
PS C:\Users\CHECH\OneDrive\Documentos\TEC\PruebasSoftware\EjerciciosPrácticos> python ComputeStatistics.py TC2.txt
sum 495800.00
count 1977.00
mean 250.78
median 247.00
mode 230.00
sd 144.21
variance 20795.89
Execution Time 0.014606714248657227s
```



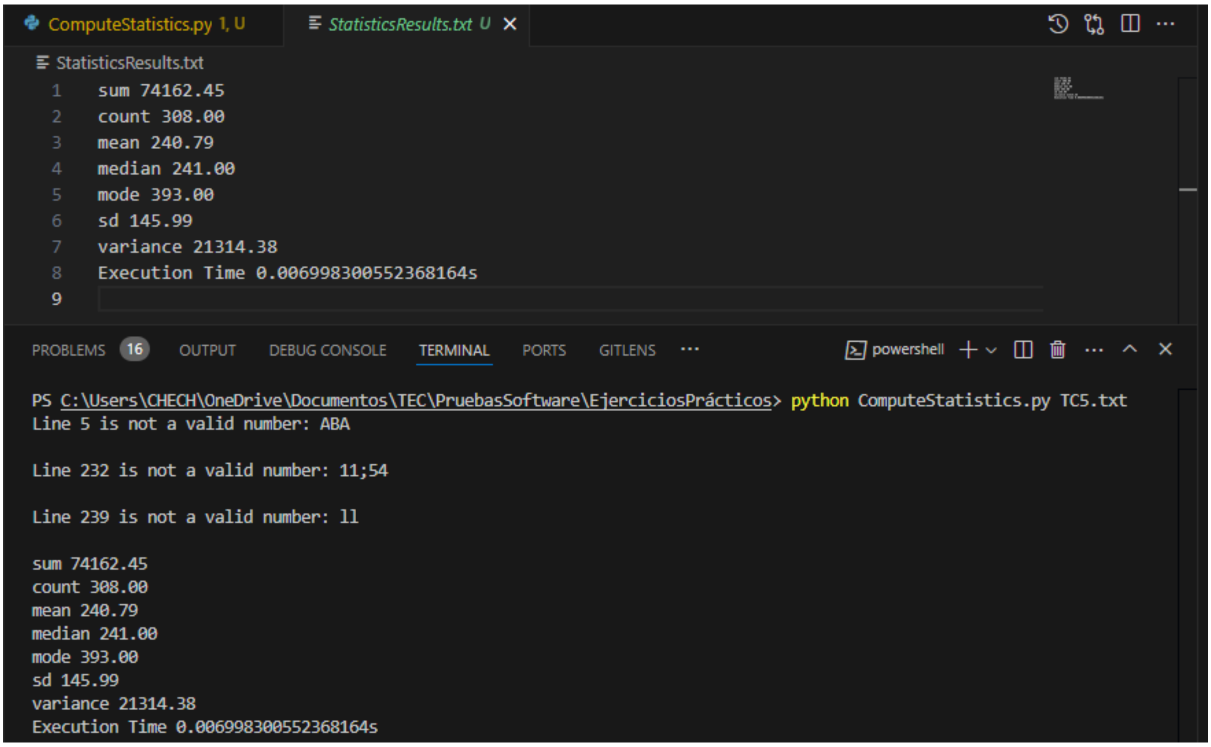
```
sum 3153175.00
count 12624.00
mean 249.78
median 249.00
mode 94.00
sd 145.32
variance 21118.95
Execution Time 0.07686853408813477s
```

```
PS C:\Users\CHECH\OneDrive\Documentos\TEC\PruebasSoftware\EjerciciosPrácticos> python ComputeStatistics.py TC3.txt
sum 3153175.00
count 12624.00
mean 249.78
median 249.00
mode 94.00
sd 145.32
variance 21118.95
Execution Time 0.07686853408813477s
```



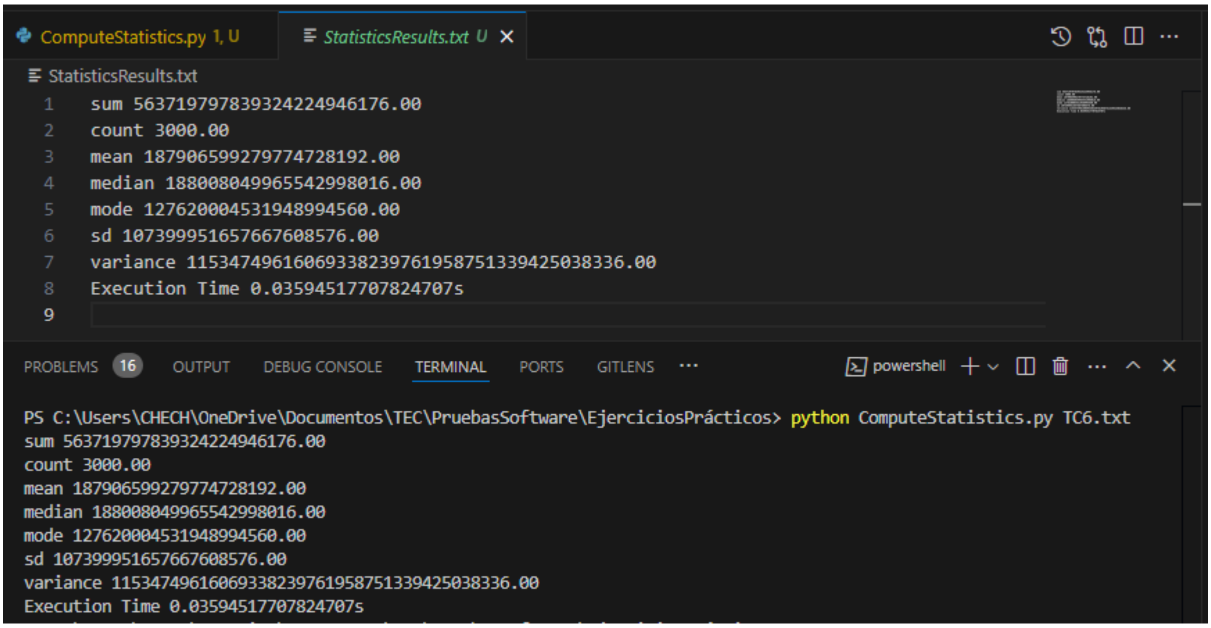
```
sum 1881009.75
count 12624.00
mean 149.00
median 147.75
mode 123.75
sd 130.42
variance 17009.27
Execution Time 0.089996337890625s
```

```
PS C:\Users\CHECH\OneDrive\Documentos\TEC\PruebasSoftware\EjerciciosPrácticos> python ComputeStatistics.py TC4.txt
sum 1881009.75
count 12624.00
mean 149.00
median 147.75
mode 123.75
sd 130.42
variance 17009.27
Execution Time 0.089996337890625s
```



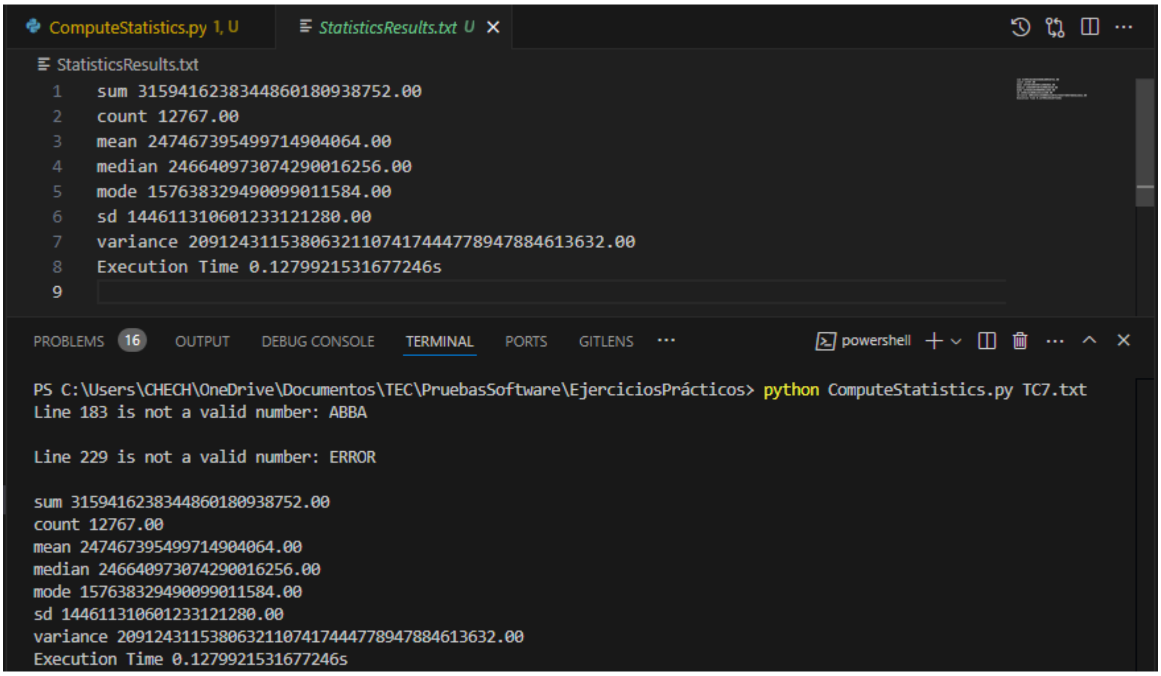
```
sum 74162.45
count 308.00
mean 240.79
median 241.00
mode 393.00
sd 145.99
variance 21314.38
Execution Time 0.006998300552368164s

```

```
PS C:\Users\CHECH\OneDrive\Documentos\TEC\PruebasSoftware\EjerciciosPrácticos> python ComputeStatistics.py TC5.txt
Line 5 is not a valid number: ABA

Line 232 is not a valid number: 11;54

Line 239 is not a valid number: 1l

sum 74162.45
count 308.00
mean 240.79
median 241.00
mode 393.00
sd 145.99
variance 21314.38
Execution Time 0.006998300552368164s
```



```
sum 5637197978393242224946176.00
count 3000.00
mean 1879065992797774728192.00
median 188008049965542998016.00
mode 127620004531948994560.00
sd 107399951657667608576.00
variance 11534749616069338239761958751339425038336.00
Execution Time 0.03594517707824707s

```

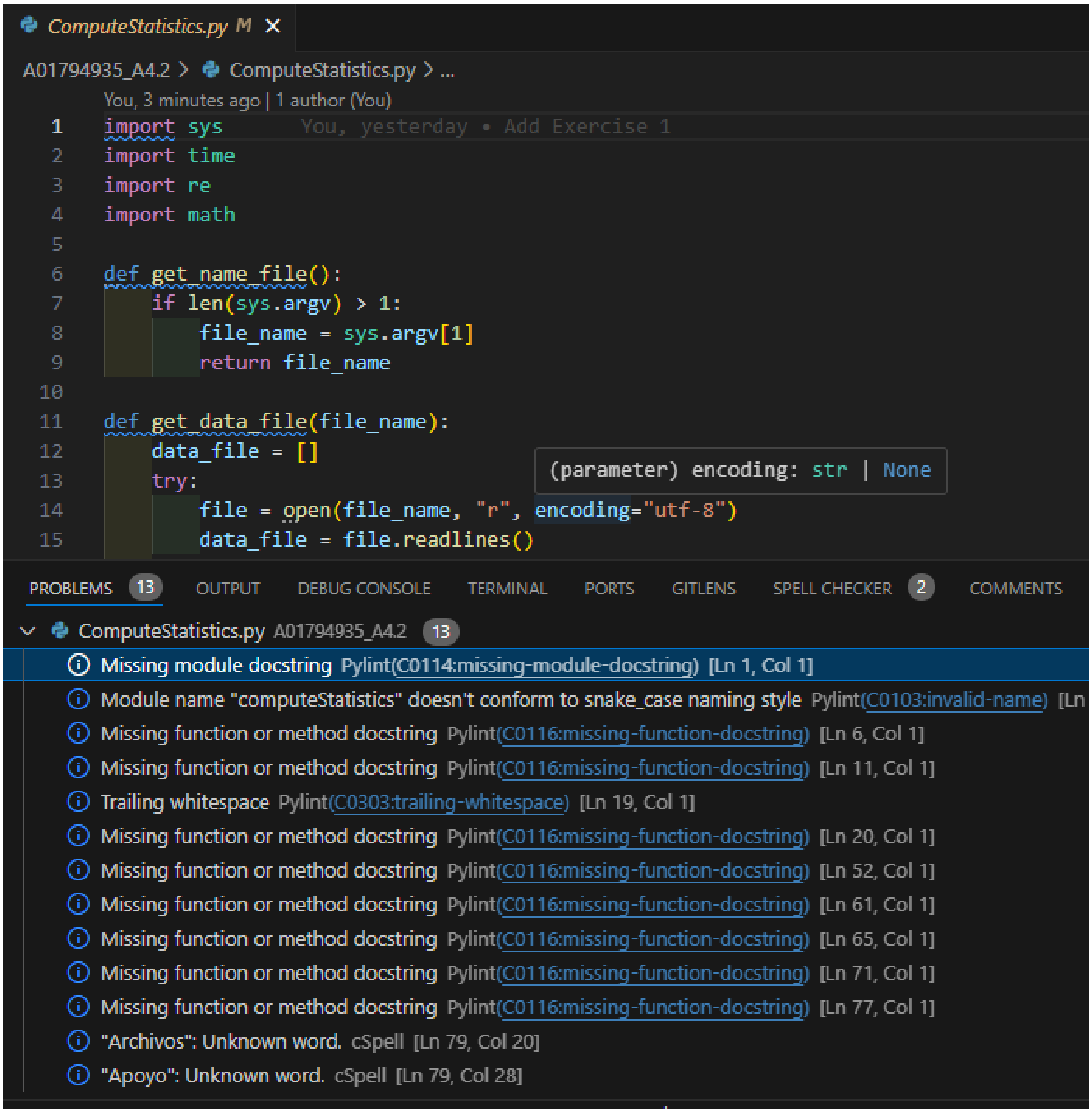
```
PS C:\Users\CHECH\OneDrive\Documentos\TEC\PruebasSoftware\EjerciciosPrácticos> python ComputeStatistics.py TC6.txt
sum 5637197978393242224946176.00
count 3000.00
mean 1879065992797774728192.00
median 188008049965542998016.00
mode 127620004531948994560.00
sd 107399951657667608576.00
variance 11534749616069338239761958751339425038336.00
Execution Time 0.03594517707824707s
```



```
sum 31594162383448601809387520.00
count 12767.00
mean 2474673954997149040640.00
median 2466409730742900162560.00
mode 1576383294900990115840.00
sd 1446113106012331212800.00
variance 20912431153806321107417444778947884613632.00
Execution Time 0.1279921531677246s
```

```
PS C:\Users\CHECH\OneDrive\Documentos\TEC\PruebasSoftware\EjerciciosPrácticos> python ComputeStatistics.py TC7.txt
Line 183 is not a valid number: ABBA

Line 229 is not a valid number: ERROR

sum 31594162383448601809387520.00
count 12767.00
mean 2474673954997149040640.00
median 2466409730742900162560.00
mode 1576383294900990115840.00
sd 1446113106012331212800.00
variance 20912431153806321107417444778947884613632.00
Execution Time 0.1279921531677246s
```

- Pylint



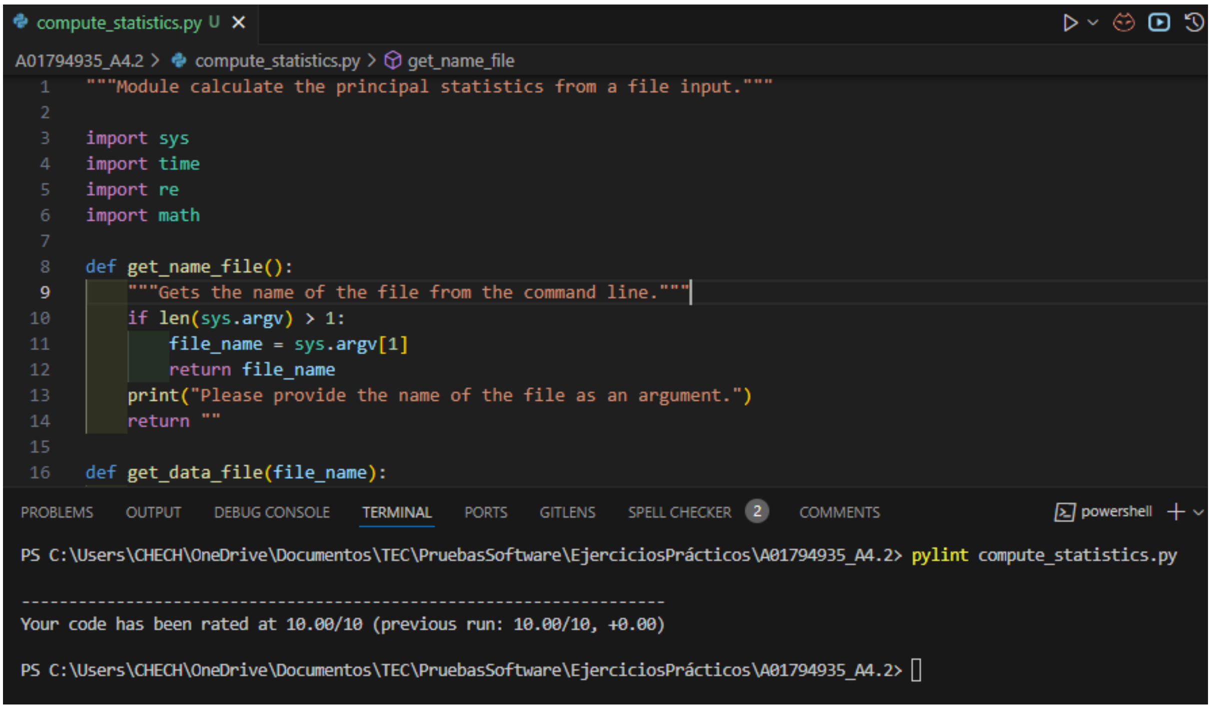
ComputeStatistics.py M ×
A01794935_A4.2 > ComputeStatistics.py > ...
You, 3 minutes ago | 1 author (You)

```
import sys        You, yesterday • Add Exercise 1
import time
import re
import math

def get_name_file():
    if len(sys.argv) > 1:
        file_name = sys.argv[1]
        return file_name

def get_data_file(file_name):
    data_file = []                        (parameter) encoding: str | None
    try:
        file = open(file_name, "r", encoding="utf-8")
        data_file = file.readlines()
```

PROBLEMS 13
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
GITLENS
SPELL CHECKER 2
COMMENTS
ComputeStatistics.py A01794935_A4.2 13

```
ⓘ Missing module docstring Pylint(C0114:missing-module-docstring) [Ln 1, Col 1]
ⓘ Module name "computeStatistics" doesn't conform to snake_case naming style Pylint(C0103:invalid-name) [Ln
ⓘ Missing function or method docstring Pylint(C0116:missing-function-docstring) [Ln 6, Col 1]
ⓘ Missing function or method docstring Pylint(C0116:missing-function-docstring) [Ln 11, Col 1]
ⓘ Trailing whitespace Pylint(C0303:trailing-whitespace) [Ln 19, Col 1]
ⓘ Missing function or method docstring Pylint(C0116:missing-function-docstring) [Ln 20, Col 1]
ⓘ Missing function or method docstring Pylint(C0116:missing-function-docstring) [Ln 52, Col 1]
ⓘ Missing function or method docstring Pylint(C0116:missing-function-docstring) [Ln 61, Col 1]
ⓘ Missing function or method docstring Pylint(C0116:missing-function-docstring) [Ln 65, Col 1]
ⓘ Missing function or method docstring Pylint(C0116:missing-function-docstring) [Ln 71, Col 1]
ⓘ Missing function or method docstring Pylint(C0116:missing-function-docstring) [Ln 77, Col 1]
ⓘ "Archivos": Unknown word. cSpell [Ln 79, Col 20]
ⓘ "Apoyo": Unknown word. cSpell [Ln 79, Col 28]
```

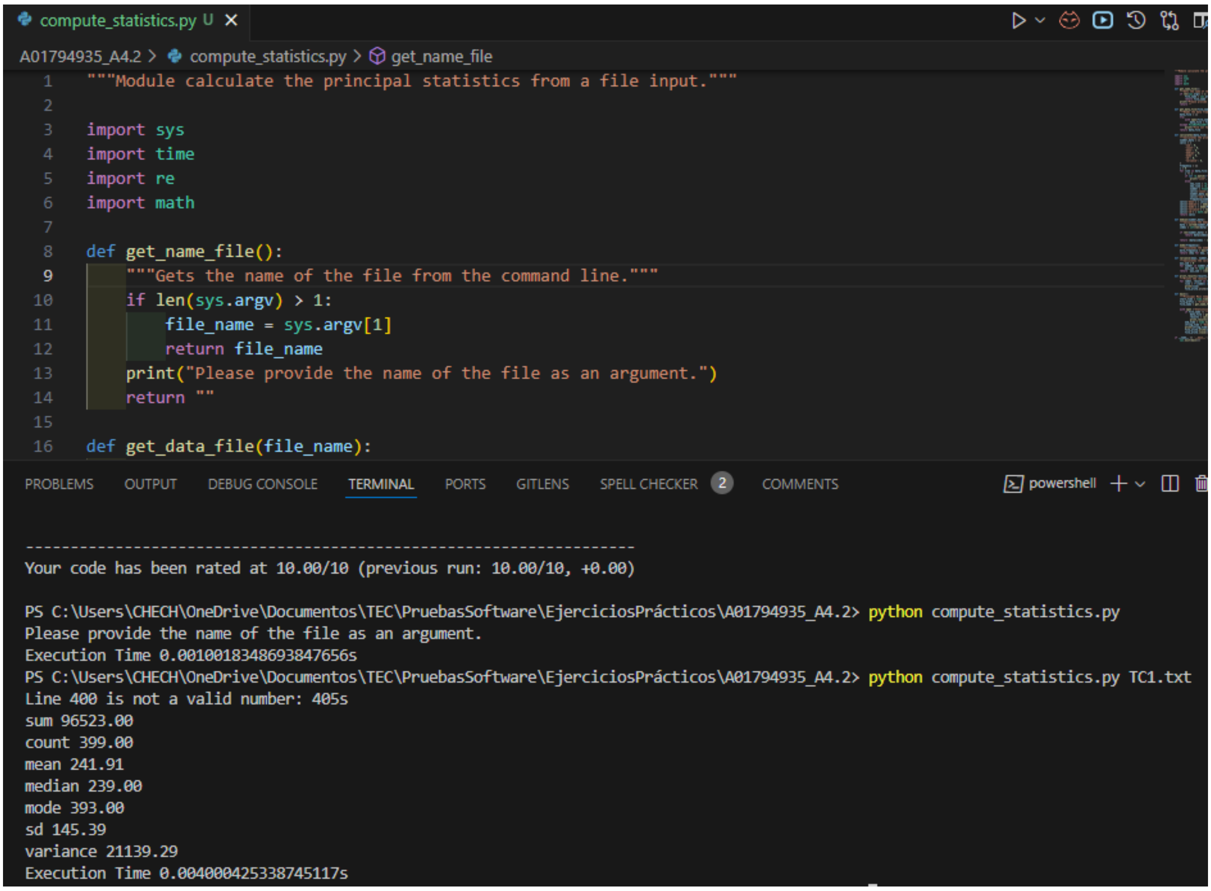
It was necessary to change the file name because don't compliance with the snake_case naming style.



```python
"""Module calculate the principal statistics from a file input."""

import sys
import time
import re
import math

def get_name_file():
    """Gets the name of the file from the command line."""
    if len(sys.argv) > 1:
        file_name = sys.argv[1]
        return file_name
    print("Please provide the name of the file as an argument.")
    return ""

def get_data_file(file_name):
```

```
PS C:\Users\CHECH\OneDrive\Documentos\TEC\PruebasSoftware\EjerciciosPrácticos\A01794935_A4.2> pylint compute_statistics.py

-------------------------------------------------------------------
Your code has been rated at 10.00/10 (previous run: 10.00/10, +0.00)

PS C:\Users\CHECH\OneDrive\Documentos\TEC\PruebasSoftware\EjerciciosPrácticos\A01794935_A4.2> 
```

The code is executed successfully.



```python
"""Module calculate the principal statistics from a file input."""

import sys
import time
import re
import math

def get_name_file():
    """Gets the name of the file from the command line."""
    if len(sys.argv) > 1:
        file_name = sys.argv[1]
        return file_name
    print("Please provide the name of the file as an argument.")
    return ""

def get_data_file(file_name):
```

```
-------------------------------------------------------------------
Your code has been rated at 10.00/10 (previous run: 10.00/10, +0.00)

PS C:\Users\CHECH\OneDrive\Documentos\TEC\PruebasSoftware\EjerciciosPrácticos\A01794935_A4.2> python compute_statistics.py
Please provide the name of the file as an argument.
Execution Time 0.0010018348693847656s
PS C:\Users\CHECH\OneDrive\Documentos\TEC\PruebasSoftware\EjerciciosPrácticos\A01794935_A4.2> python compute_statistics.py TC1.txt
Line 400 is not a valid number: 405s
sum 96523.00
count 399.00
mean 241.91
median 239.00
mode 393.00
sd 145.39
variance 21139.29
Execution Time 0.004000425338745117s
```

## Exercise #2: Converter

☒Req1. The program shall be invoked from a command line. The program shall receive a file as parameter. The file will contain a list of items (presumable numbers).

☒Req 2. The program shall convert the numbers to binary and hexadecimal base. The results shall be print on a screen and on a file named ConvertionResults.txt. All computation MUST be calculated using the basic algorithms, not functions or libraries.

☒Req 3. The program shall include the mechanism to handle invalid data in the file. Errors should be displayed in the console and the execution must continue.

☒Req 4. The name of the program shall be convertNumbers.py

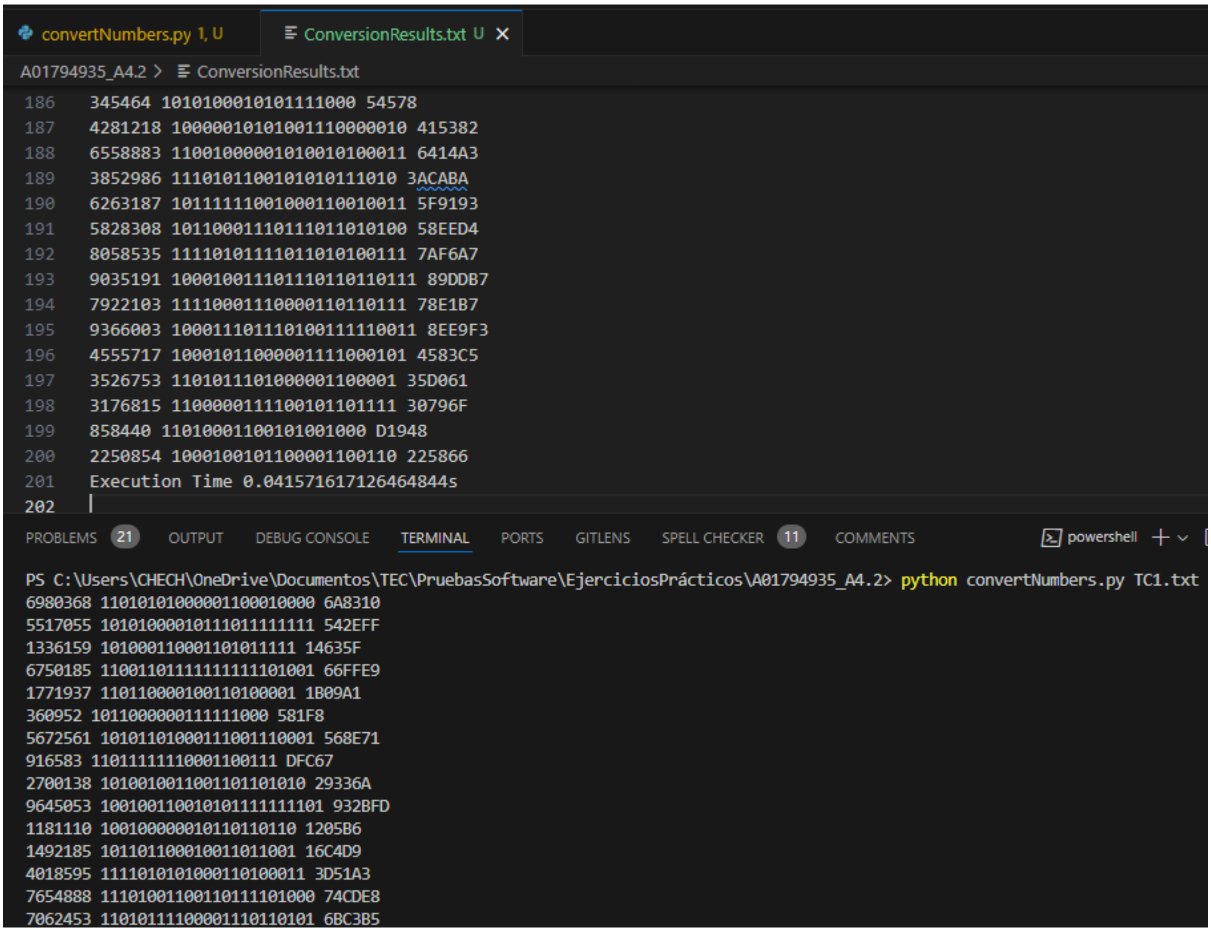
☒Req 5. The minimum format to invoke the program shall be as follows: python convertNumbers.py fileWithData.txt

☒Req 6. The program shall manage files having from hundreds of items to thousands of items.

☒Req 7. The program should include at the end of the execution the time elapsed for the execution and calculus of the data. This number shall be included in the results file and on the screen.

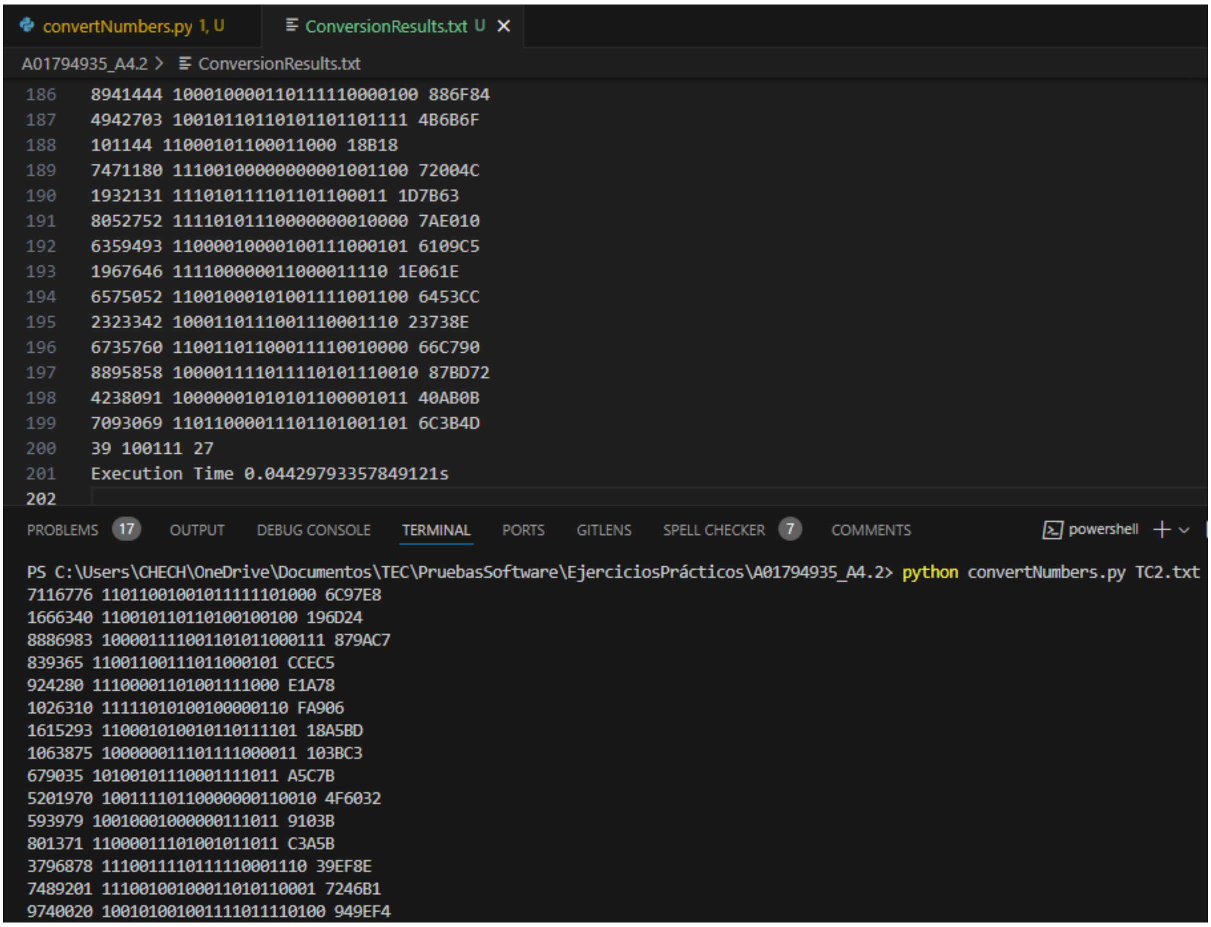
☒Req 8. Be compliant with PEP8.

- Results



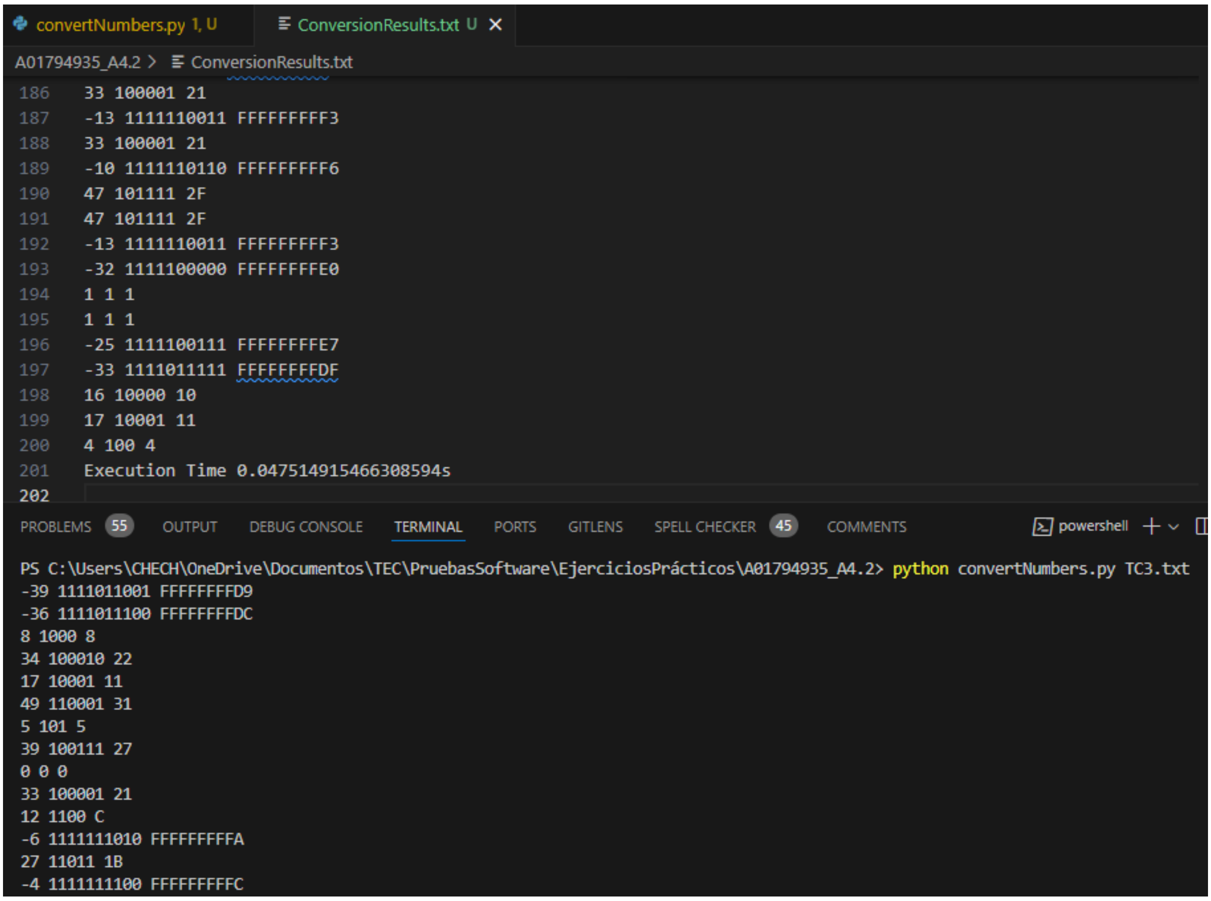
```
345464 1010100010101111000 54578
4281218 10000010101001110000010 415382
6558883 11001000001010010100011 6414A3
3852986 1110101100101010111010 3ACABA
6263187 10111111001000110010011 5F9193
5828308 10110001110111011010100 58EED4
8058535 11110101111011010100111 7AF6A7
9035191 100010011101110110110111 89DDB7
7922103 11110001110000110110111 78E1B7
9366003 100011101110100111110011 8EE9F3
4555717 10001011000001111000101 4583C5
3526753 1101011101000001100001 35D061
3176815 1100000111100101101111 30796F
858440 11010001100101001000 D1948
2250854 1000100101100001100110 225866
Execution Time 0.041571617126464844s
```

```
PS C:\Users\CHECH\OneDrive\Documentos\TEC\PruebasSoftware\EjerciciosPrácticos\A01794935_A4.2> python convertNumbers.py TC1.txt
6980368 11010101000001100010000 6A8310
5517055 10101000010111011111111 542EFF
1336159 101000110001101011111 14635F
6750185 11001101111111111101001 66FFE9
1771937 110110000100110100001 1B09A1
360952 1011000000111111000 581F8
5672561 10101101000111001110001 568E71
916583 11011111110001100111 DFC67
2700138 1010010011001101101010 29336A
9645053 100100110010101111111101 932BFD
1181110 100100000010110110110 1205B6
1492185 101101100010011011001 16C4D9
4018595 1111010101000110100011 3D51A3
7654888 11101001100110111101000 74CDE8
7062453 11010111100001110110101 6BC3B5
```



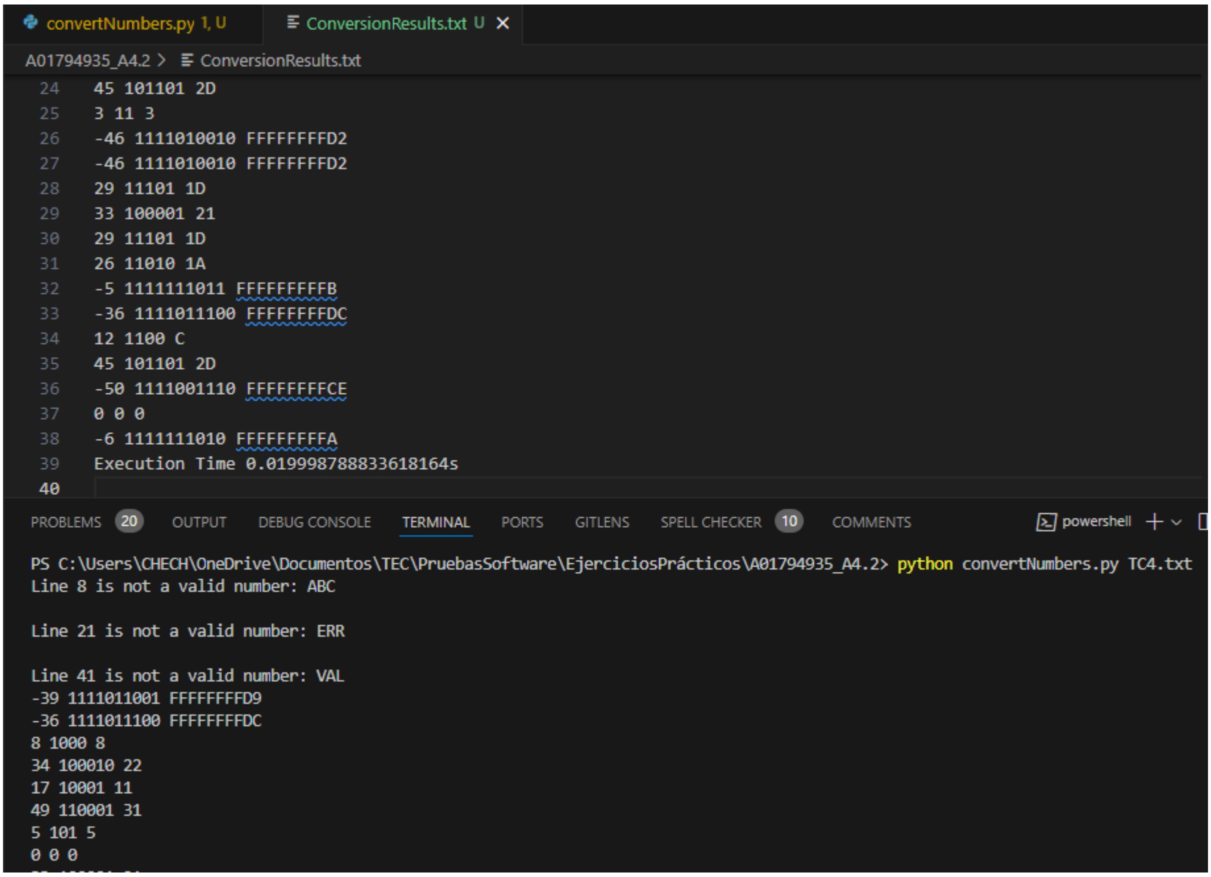
```
8941444 1000100001101111110000100 886F84
4942703 10010110110101101101111 4B6B6F
101144 11000101100011000 18B18
7471180 11100100000000001001100 72004C
1932131 111010111101101100011 1D7B63
8052752 11110101110000000010000 7AE010
6359493 11000010000100111000101 6109C5
1967646 111100000011000011110 1E061E
6575052 11001000101001111001100 6453CC
2323342 1000110111001110001110 23738E
6735760 11001101100011110010000 66C790
8895858 100001111011110101110010 87BD72
4238091 10000001010101100001011 40AB0B
7093069 11011000011101101001101 6C3B4D
39 100111 27
Execution Time 0.04429793357849121s
```

```
PS C:\Users\CHECH\OneDrive\Documentos\TEC\PruebasSoftware\EjerciciosPrácticos\A01794935_A4.2> python convertNumbers.py TC2.txt
7116776 11011001001011111101000 6C97E8
1666340 110010110110100100100 196D24
8886983 100001111001101011000111 879AC7
839365 11001100111011000101 CCEC5
924280 11100001101001111000 E1A78
1026310 11111010100100000110 FA906
1615293 110001010010110111101 18A5BD
1063875 100000011101111000011 103BC3
679035 10100101110001111011 A5C7B
5201970 10011110110000000110010 4F6032
593979 10010001000000111011 9103B
801371 11000011101001011011 C3A5B
3796878 1110011110111110001110 39EF8E
7489201 11100100100011010110001 7246B1
9740020 100101001001111011110100 949EF4
```



```
33 100001 21
-13 1111110011 FFFFFFFFF3
33 100001 21
-10 1111110110 FFFFFFFFF6
47 101111 2F
47 101111 2F
-13 1111110011 FFFFFFFFF3
-32 1111100000 FFFFFFFFE0
1 1 1
1 1 1
-25 1111100111 FFFFFFFFE7
-33 1111011111 FFFFFFFFDF
16 10000 10
17 10001 11
4 100 4
Execution Time 0.047514915466308594s
```

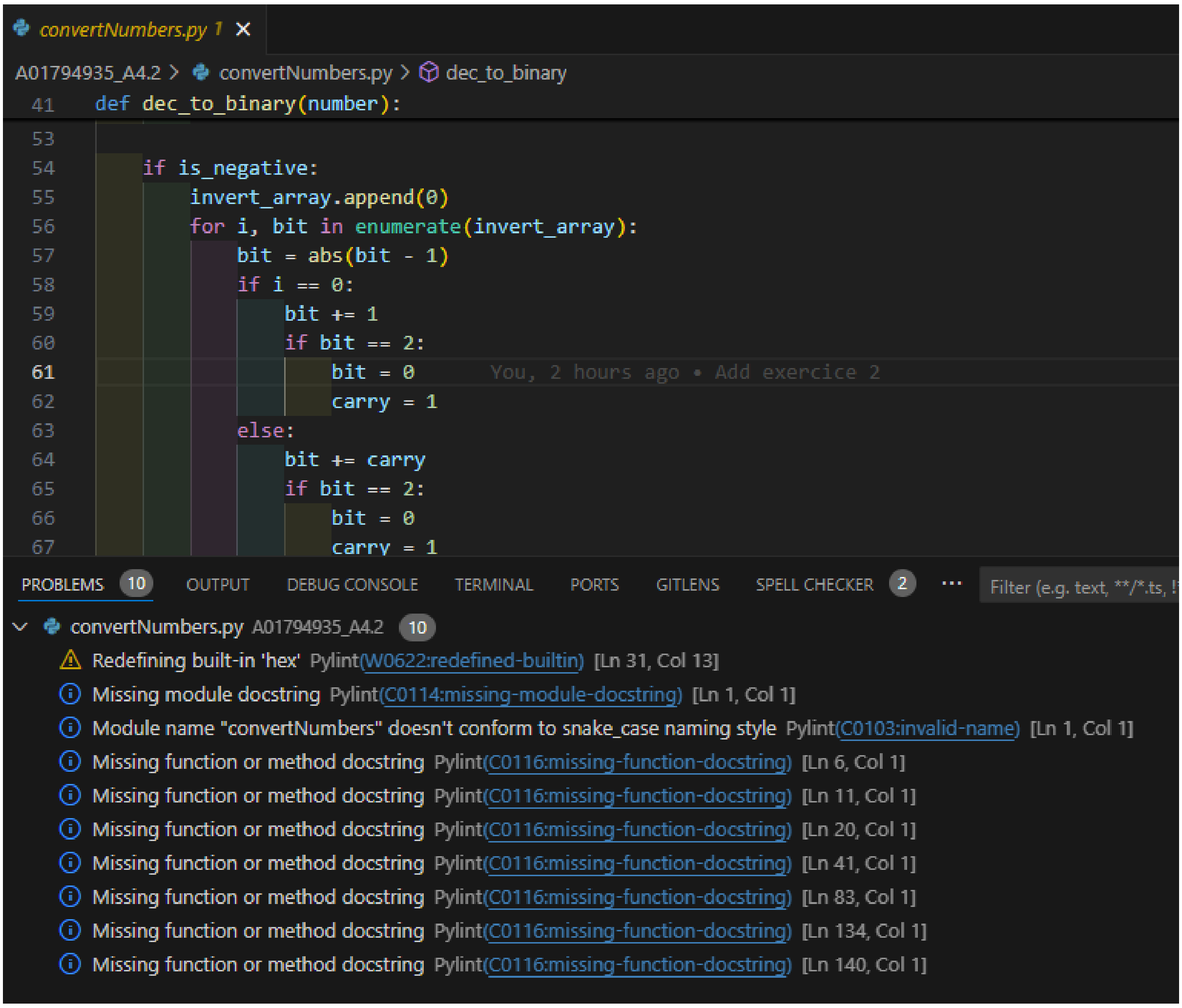
```
PS C:\Users\CHECH\OneDrive\Documentos\TEC\PruebasSoftware\EjerciciosPrácticos\A01794935_A4.2> python convertNumbers.py TC3.txt
-39 1111011001 FFFFFFFFD9
-36 1111011100 FFFFFFFFDC
8 1000 8
34 100010 22
17 10001 11
49 110001 31
5 101 5
39 100111 27
0 0 0
33 100001 21
12 1100 C
-6 1111111010 FFFFFFFFFA
27 11011 1B
-4 1111111100 FFFFFFFFFC
```



```
45 101101 2D
3 11 3
-46 1111010010 FFFFFFFFD2
-46 1111010010 FFFFFFFFD2
29 11101 1D
33 100001 21
29 11101 1D
26 11010 1A
-5 1111111011 FFFFFFFFFB
-36 1111011100 FFFFFFFFDC
12 1100 C
45 101101 2D
-50 1111001110 FFFFFFFFCE
0 0 0
-6 1111111010 FFFFFFFFFA
Execution Time 0.019998788833618164s
```

```
PS C:\Users\CHECH\OneDrive\Documentos\TEC\PruebasSoftware\EjerciciosPrácticos\A01794935_A4.2> python convertNumbers.py TC4.txt
Line 8 is not a valid number: ABC

Line 21 is not a valid number: ERR

Line 41 is not a valid number: VAL
-39 1111011001 FFFFFFFFD9
-36 1111011100 FFFFFFFFDC
8 1000 8
34 100010 22
17 10001 11
49 110001 31
5 101 5
0 0 0
```

- Pylint



convertNumbers.py 1 ×
A01794935_A4.2 > convertNumbers.py > dec_to_binary

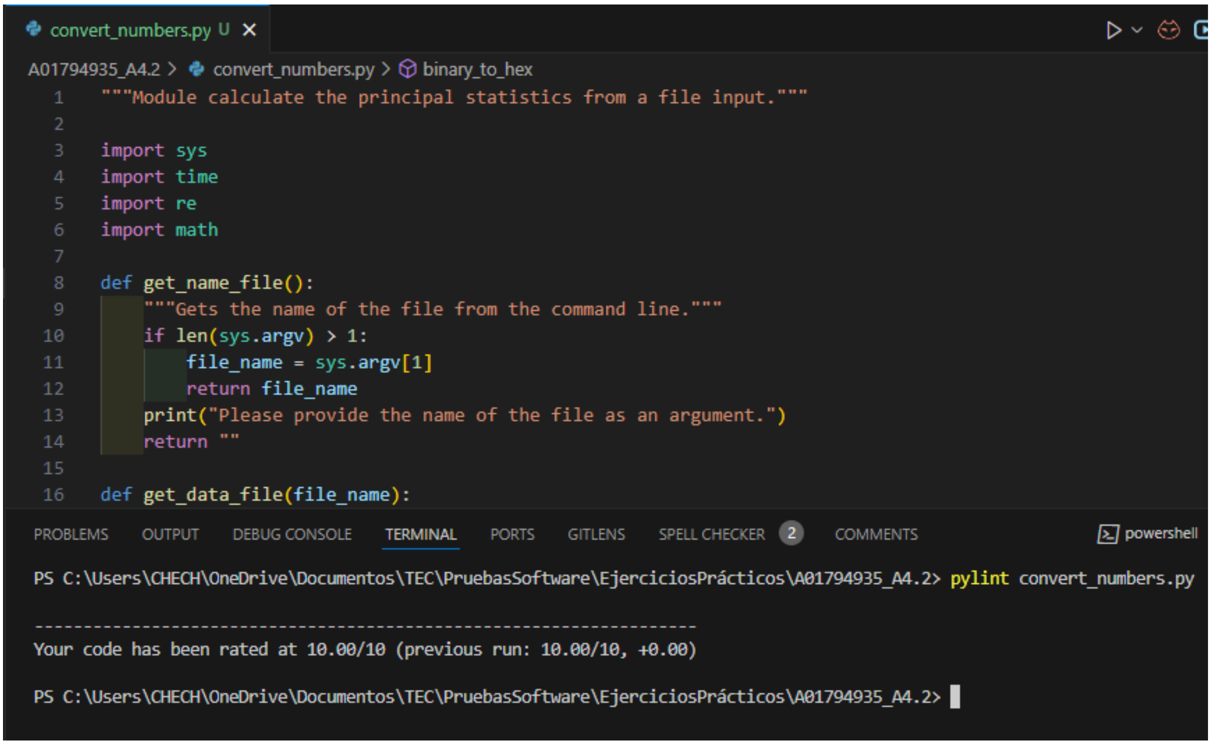
```
def dec_to_binary(number):

    if is_negative:
        invert_array.append(0)
        for i, bit in enumerate(invert_array):
            bit = abs(bit - 1)
            if i == 0:
                bit += 1
                if bit == 2:
                    bit = 0        You, 2 hours ago • Add exercice 2
                    carry = 1
            else:
                bit += carry
                if bit == 2:
                    bit = 0
                    carry = 1
```

PROBLEMS 10 OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS SPELL CHECKER 2 ··· Filter (e.g. text, **/*.ts, !

convertNumbers.py A01794935_A4.2 10

⚠ Redefining built-in 'hex' Pylint(W0622:redefined-builtin) [Ln 31, Col 13]
ⓘ Missing module docstring Pylint(C0114:missing-module-docstring) [Ln 1, Col 1]
ⓘ Module name "convertNumbers" doesn't conform to snake_case naming style Pylint(C0103:invalid-name) [Ln 1, Col 1]
ⓘ Missing function or method docstring Pylint(C0116:missing-function-docstring) [Ln 6, Col 1]
ⓘ Missing function or method docstring Pylint(C0116:missing-function-docstring) [Ln 11, Col 1]
ⓘ Missing function or method docstring Pylint(C0116:missing-function-docstring) [Ln 20, Col 1]
ⓘ Missing function or method docstring Pylint(C0116:missing-function-docstring) [Ln 41, Col 1]
ⓘ Missing function or method docstring Pylint(C0116:missing-function-docstring) [Ln 83, Col 1]
ⓘ Missing function or method docstring Pylint(C0116:missing-function-docstring) [Ln 134, Col 1]
ⓘ Missing function or method docstring Pylint(C0116:missing-function-docstring) [Ln 140, Col 1]

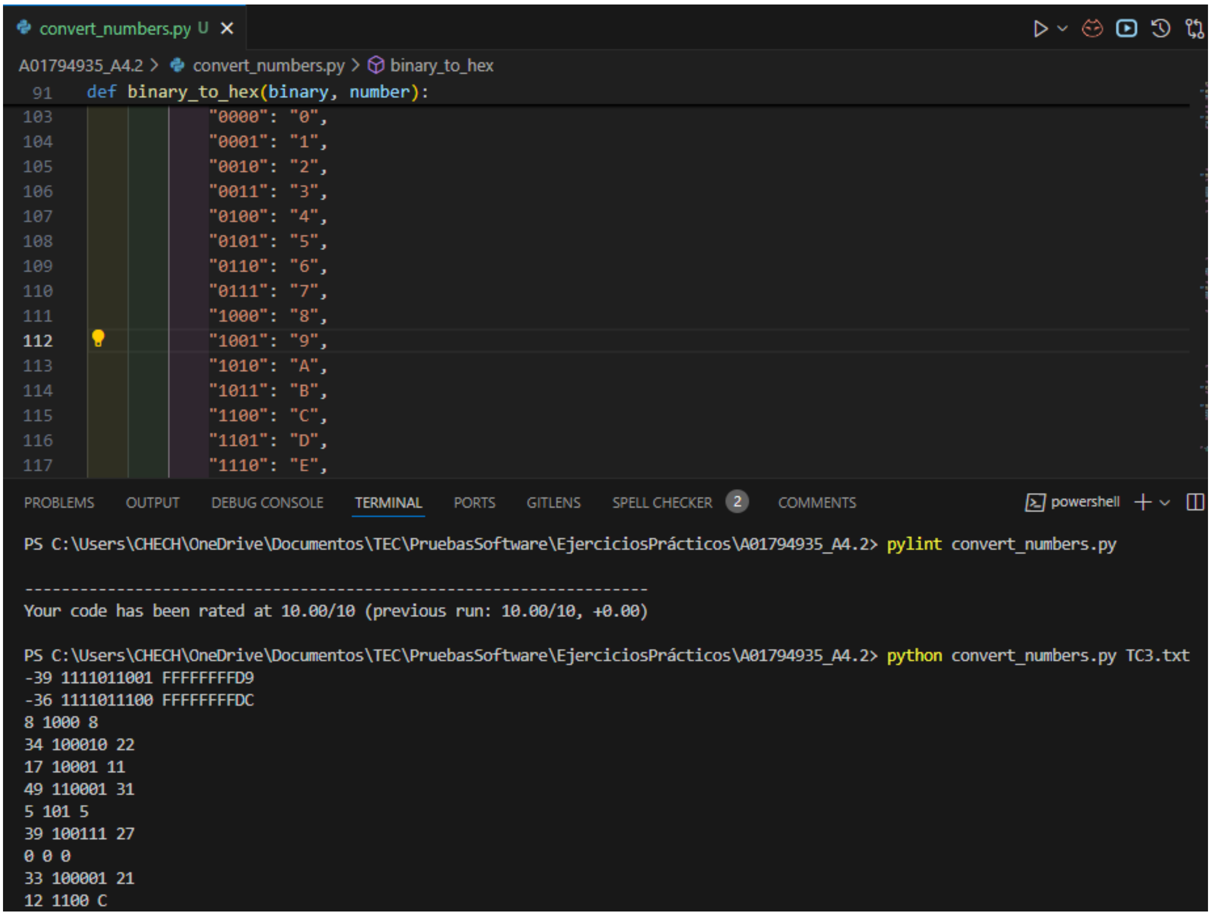
It was necessary to change the file name because don't compliance with the snake_case naming style.


convert_numbers.py U ×
A01794935_A4.2 > convert_numbers.py > binary_to_hex
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS SPELL CHECKER 2 COMMENTS
powershell

```python
"""Module calculate the principal statistics from a file input."""

import sys
import time
import re
import math

def get_name_file():
    """Gets the name of the file from the command line."""
    if len(sys.argv) > 1:
        file_name = sys.argv[1]
        return file_name
    print("Please provide the name of the file as an argument.")
    return ""

def get_data_file(file_name):
```

```
PS C:\Users\CHECH\OneDrive\Documentos\TEC\PruebasSoftware\EjerciciosPrácticos\A01794935_A4.2> pylint convert_numbers.py

-------------------------------------------------------------------
Your code has been rated at 10.00/10 (previous run: 10.00/10, +0.00)

PS C:\Users\CHECH\OneDrive\Documentos\TEC\PruebasSoftware\EjerciciosPrácticos\A01794935_A4.2> 
```

The code is executed successfully.


convert_numbers.py U ×
A01794935_A4.2 > convert_numbers.py > binary_to_hex
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS SPELL CHECKER 2 COMMENTS
powershell

```python
def binary_to_hex(binary, number):
            "0000": "0",
            "0001": "1",
            "0010": "2",
            "0011": "3",
            "0100": "4",
            "0101": "5",
            "0110": "6",
            "0111": "7",
            "1000": "8",
            "1001": "9",
            "1010": "A",
            "1011": "B",
            "1100": "C",
            "1101": "D",
            "1110": "E",
```

```
PS C:\Users\CHECH\OneDrive\Documentos\TEC\PruebasSoftware\EjerciciosPrácticos\A01794935_A4.2> pylint convert_numbers.py

-------------------------------------------------------------------
Your code has been rated at 10.00/10 (previous run: 10.00/10, +0.00)

PS C:\Users\CHECH\OneDrive\Documentos\TEC\PruebasSoftware\EjerciciosPrácticos\A01794935_A4.2> python convert_numbers.py TC3.txt
-39 1111011001 FFFFFFFFD9
-36 1111011100 FFFFFFFFDC
8 1000 8
34 100010 22
17 10001 11
49 110001 31
5 101 5
39 100111 27
0 0 0
33 100001 21
12 1100 C
```

## Exercise #3: Count Words

☒Req1. The program shall be invoked from a command line. The program shall receive a file as parameter. The file will contain a words (presumable between spaces).

☒Req 2. The program shall identify all distinct words and the frequency of them (how many times the word "X" appears in the file). The results shall be print on a screen and on a file named WordCountResults.txt. All computation MUST be calculated using the basic algorithms, not functions or libraries.

☒Req 3. The program shall include the mechanism to handle invalid data in the file. Errors should be displayed in the console and the execution must continue.

☒Req 4. The name of the program shall be wordCount.py

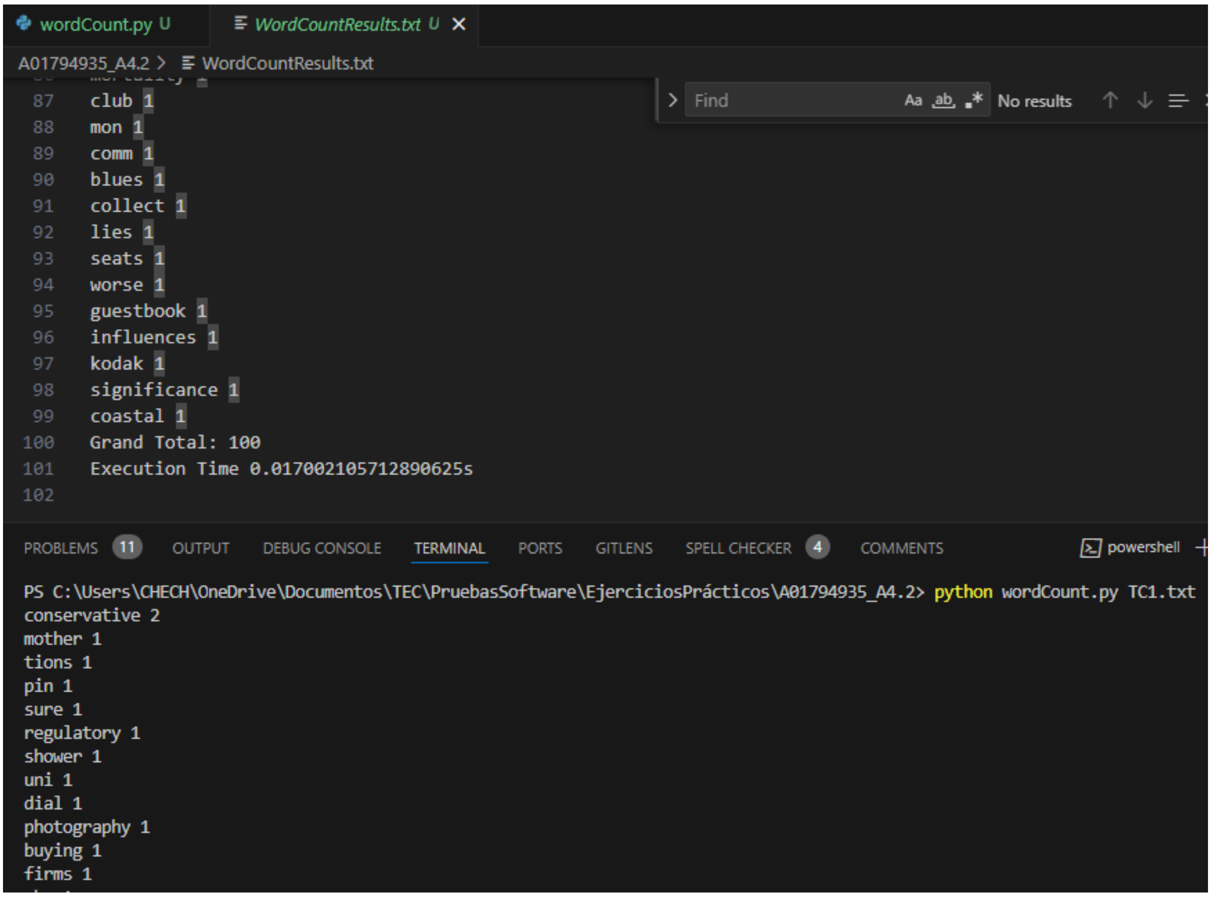
☒Req 5. The minimum format to invoke the program shall be as follows: python wordCount.py fileWithData.txt

☒Req 6. The program shall manage files having from hundreds of items to thousands of items.

☒Req 7. The program should include at the end of the execution the time elapsed for the execution and calculus of the data. This number shall be included in the results file and on the screen.

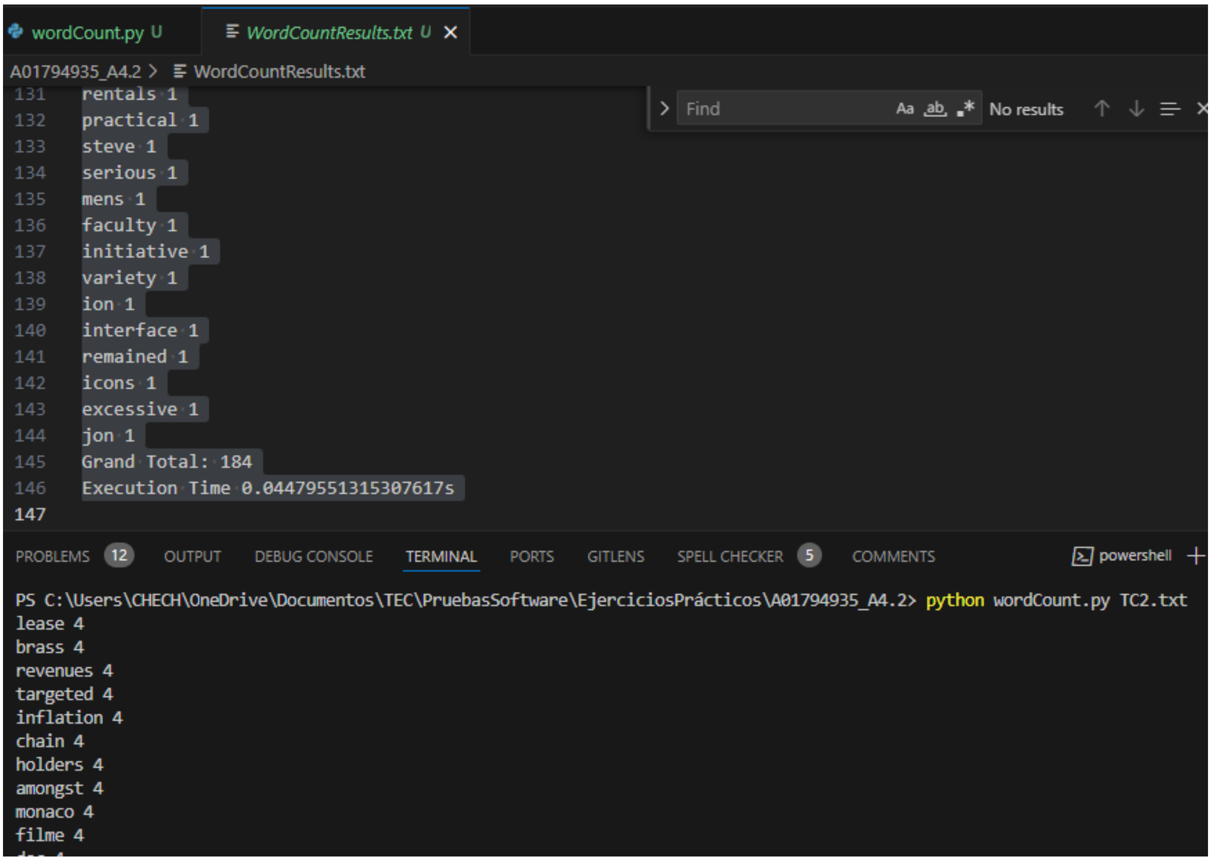
☒Req 8. Be compliant with PEP8

- ### Results



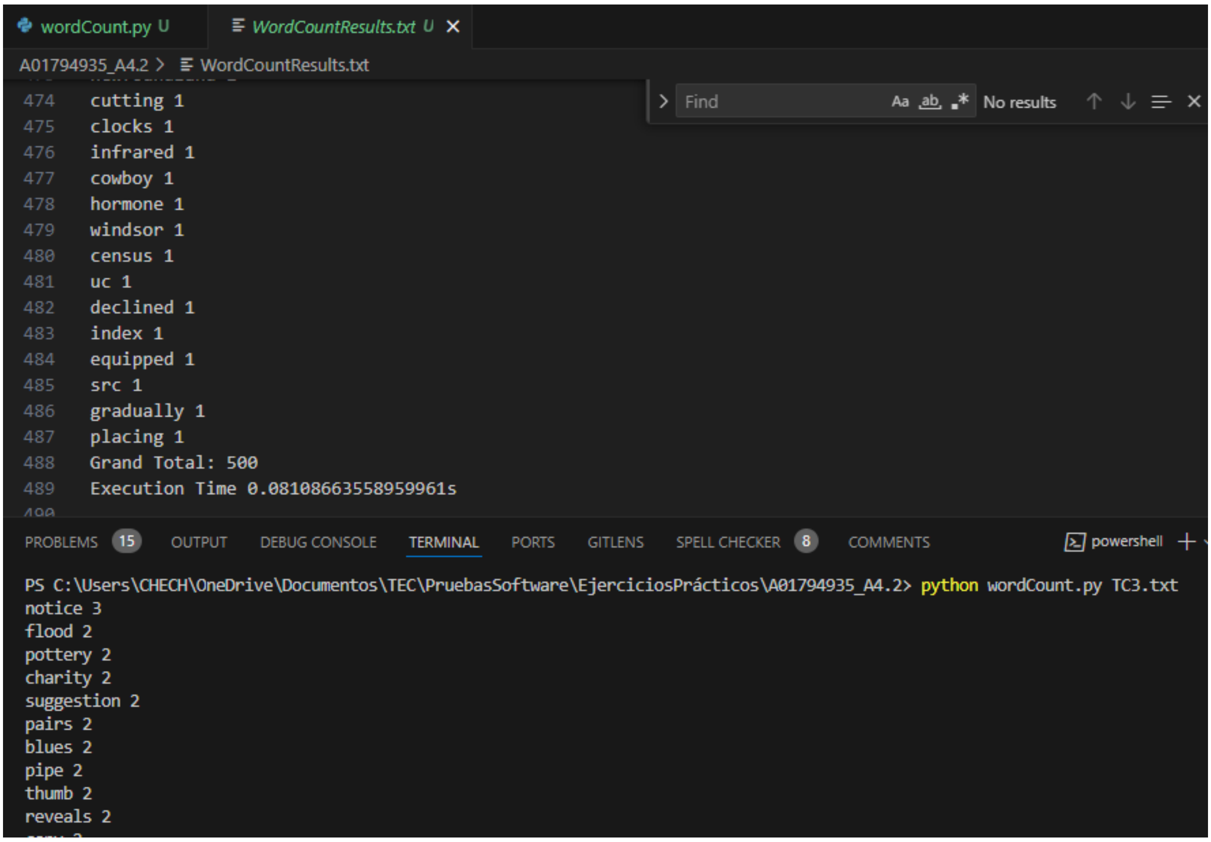
```
club 1
mon 1
comm 1
blues 1
collect 1
lies 1
seats 1
worse 1
guestbook 1
influences 1
kodak 1
significance 1
coastal 1
Grand Total: 100
Execution Time 0.017002105712890625s
```

```
PS C:\Users\CHECH\OneDrive\Documentos\TEC\PruebasSoftware\EjerciciosPrácticos\A01794935_A4.2> python wordCount.py TC1.txt
conservative 2
mother 1
tions 1
pin 1
sure 1
regulatory 1
shower 1
uni 1
dial 1
photography 1
buying 1
firms 1
```



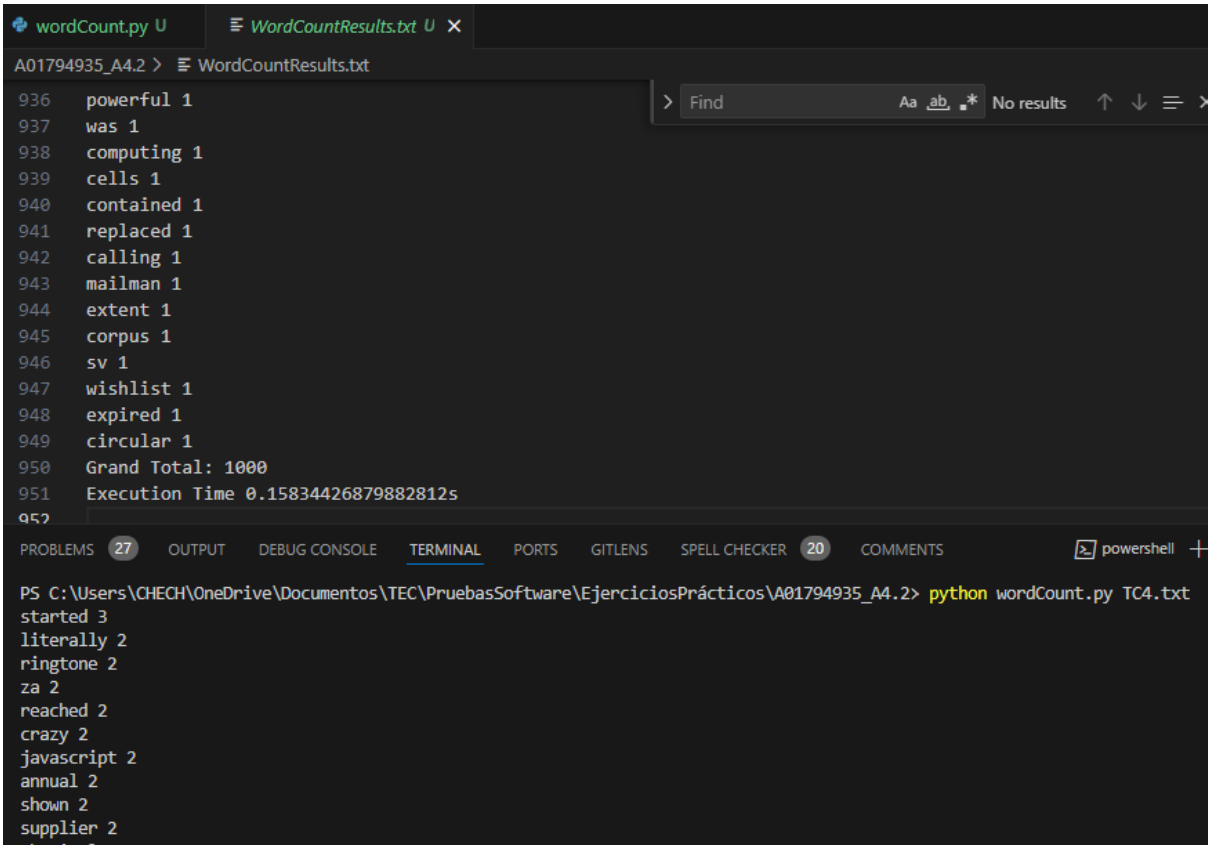
```
rentals 1
practical 1
steve 1
serious 1
mens 1
faculty 1
initiative 1
variety 1
ion 1
interface 1
remained 1
icons 1
excessive 1
jon 1
Grand Total: 184
Execution Time 0.04479551315307617s
```

```
PS C:\Users\CHECH\OneDrive\Documentos\TEC\PruebasSoftware\EjerciciosPrácticos\A01794935_A4.2> python wordCount.py TC2.txt
lease 4
brass 4
revenues 4
targeted 4
inflation 4
chain 4
holders 4
amongst 4
monaco 4
filme 4
```



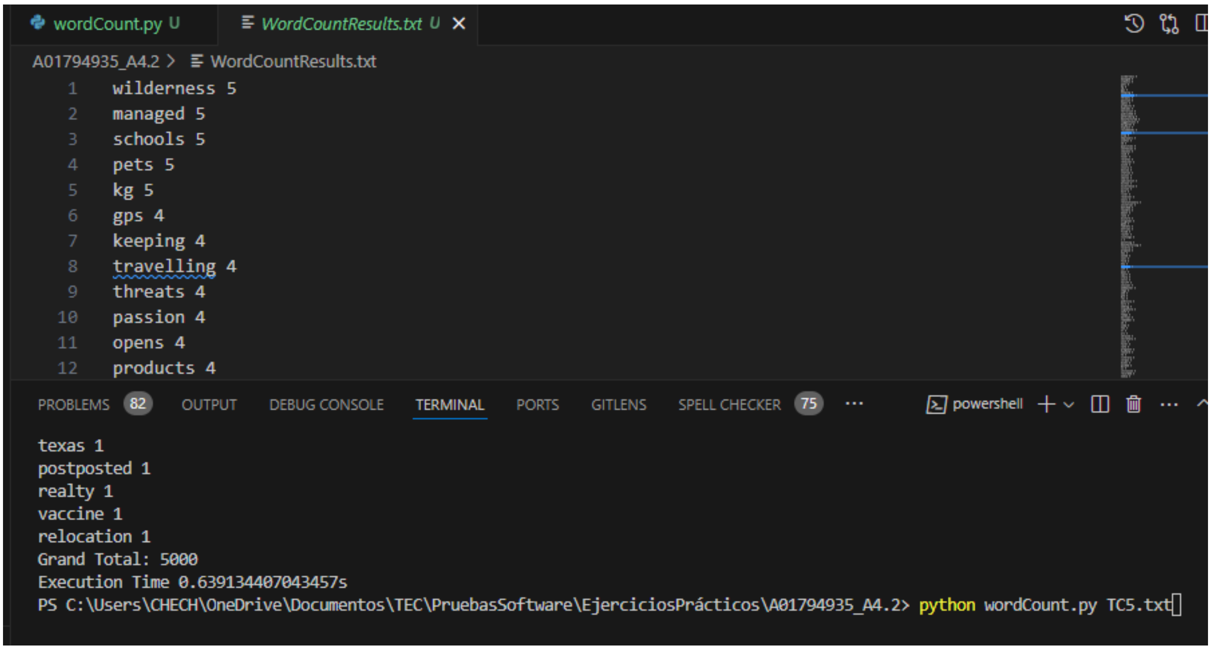
```
cutting 1
clocks 1
infrared 1
cowboy 1
hormone 1
windsor 1
census 1
uc 1
declined 1
index 1
equipped 1
src 1
gradually 1
placing 1
Grand Total: 500
Execution Time 0.08108663558959961s
```

```
PS C:\Users\CHECH\OneDrive\Documentos\TEC\PruebasSoftware\EjerciciosPrácticos\A01794935_A4.2> python wordCount.py TC3.txt
notice 3
flood 2
pottery 2
charity 2
suggestion 2
pairs 2
blues 2
pipe 2
thumb 2
reveals 2
```



```
powerful 1
was 1
computing 1
cells 1
contained 1
replaced 1
calling 1
mailman 1
extent 1
corpus 1
sv 1
wishlist 1
expired 1
circular 1
Grand Total: 1000
Execution Time 0.15834426879882812s
```

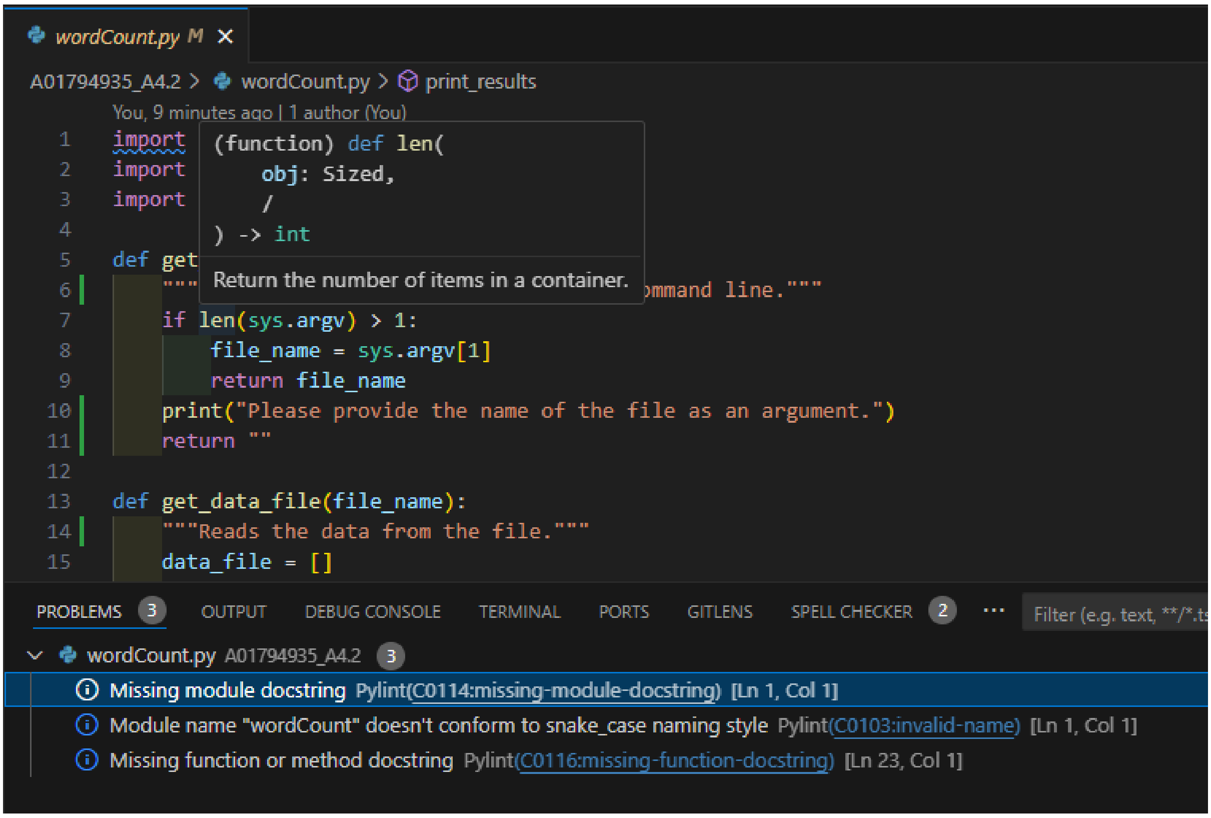
```
PS C:\Users\CHECH\OneDrive\Documentos\TEC\PruebasSoftware\EjerciciosPrácticos\A01794935_A4.2> python wordCount.py TC4.txt
started 3
literally 2
ringtone 2
za 2
reached 2
crazy 2
javascript 2
annual 2
shown 2
supplier 2
```



```
wilderness 5
managed 5
schools 5
pets 5
kg 5
gps 4
keeping 4
travelling 4
threats 4
passion 4
opens 4
products 4
```

```
texas 1
postposted 1
realty 1
vaccine 1
relocation 1
Grand Total: 5000
Execution Time 0.639134407043457s
PS C:\Users\CHECH\OneDrive\Documentos\TEC\PruebasSoftware\EjerciciosPrácticos\A01794935_A4.2> python wordCount.py TC5.txt
```

- Pylint


wordCount.py M
A01794935_A4.2 > wordCount.py > print_results
You, 9 minutes ago | 1 author (You)
PROBLEMS 3
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
GITLENS
SPELL CHECKER 2
Filter (e.g. text, **/*.ts

```
import
import
import

def get
    """                                                    ommand line."""
    if len(sys.argv) > 1:
        file_name = sys.argv[1]
        return file_name
    print("Please provide the name of the file as an argument.")
    return ""

def get_data_file(file_name):
    """Reads the data from the file."""
    data_file = []
```

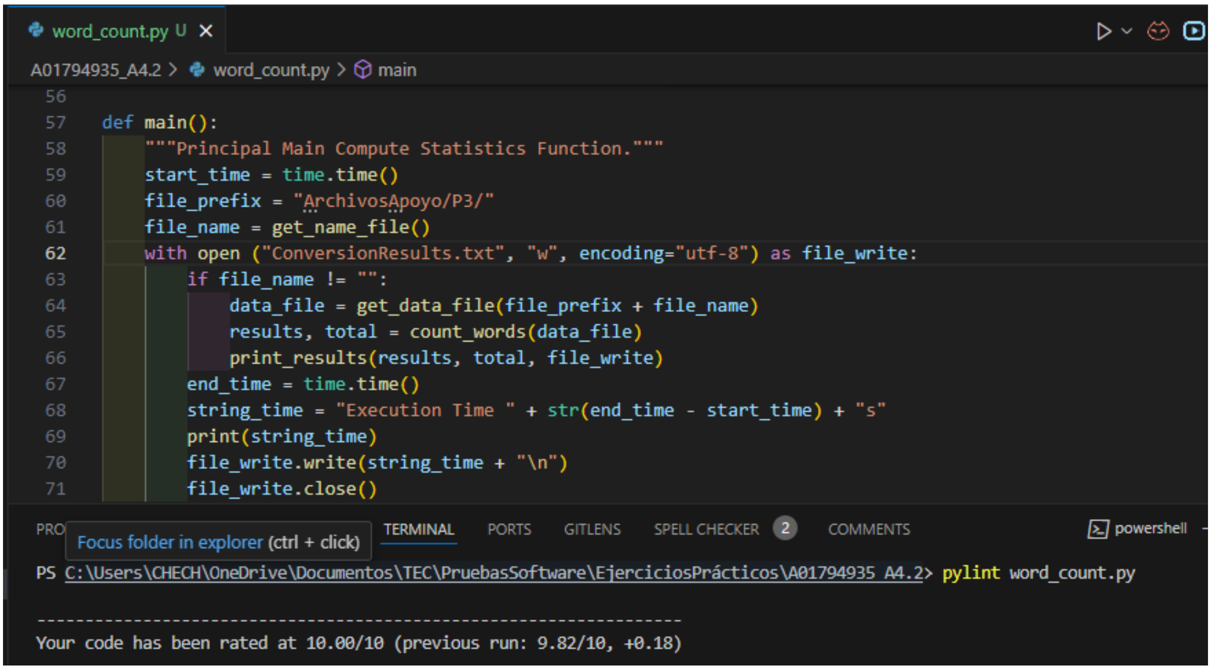
```
(function) def len(
    obj: Sized,
    /
) -> int
```

Return the number of items in a container.

wordCount.py A01794935_A4.2 3

- Missing module docstring Pylint(C0114:missing-module-docstring) [Ln 1, Col 1]
- Module name "wordCount" doesn't conform to snake_case naming style Pylint(C0103:invalid-name) [Ln 1, Col 1]
- Missing function or method docstring Pylint(C0116:missing-function-docstring) [Ln 23, Col 1]

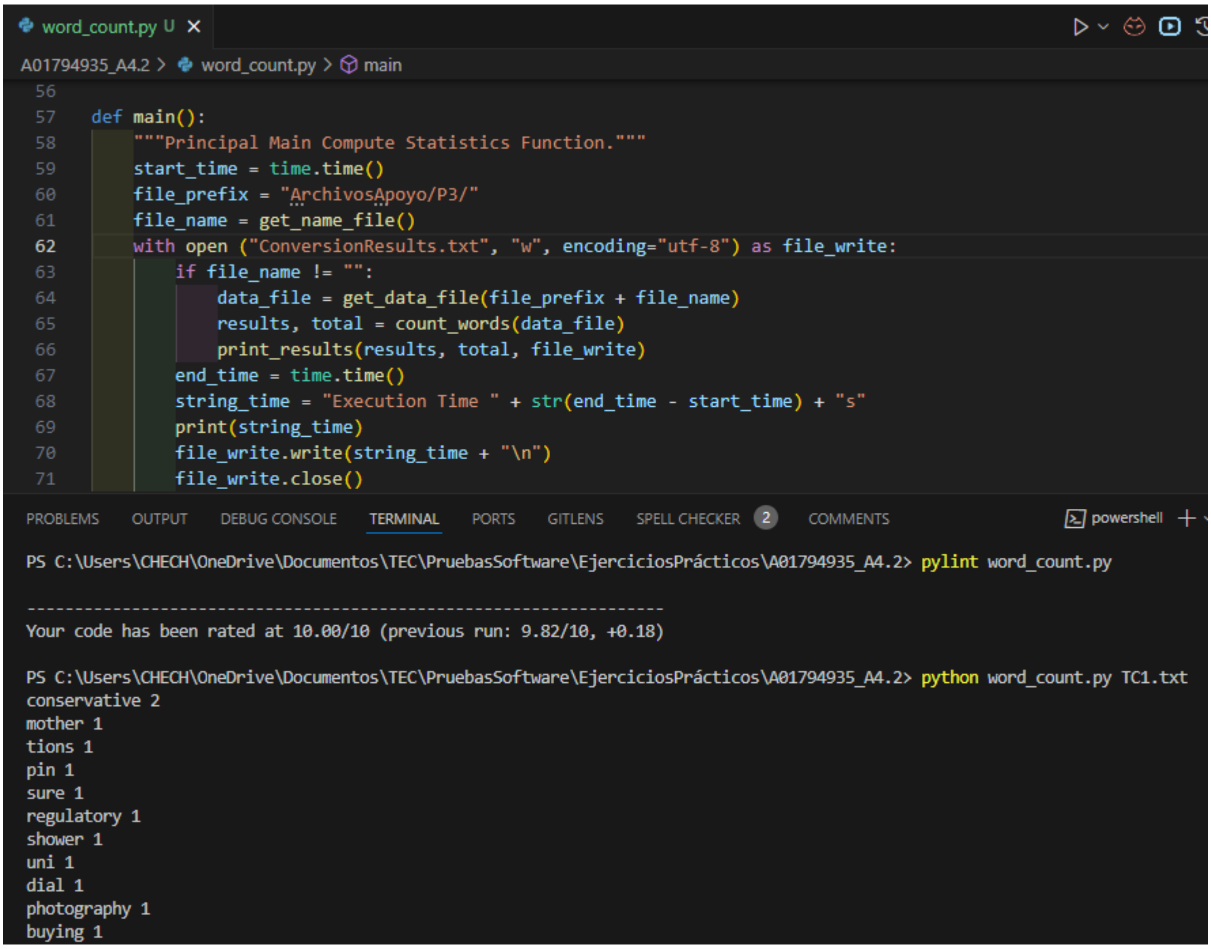
It was necessary to change the file name because don’t compliance with the snake_case naming style.


word_count.py U
A01794935_A4.2 > word_count.py > main
PRO
Focus folder in explorer (ctrl + click)
TERMINAL
PORTS
GITLENS
SPELL CHECKER 2
COMMENTS
powershell

```python

def main():
    """Principal Main Compute Statistics Function."""
    start_time = time.time()
    file_prefix = "ArchivosApoyo/P3/"
    file_name = get_name_file()
    with open ("ConversionResults.txt", "w", encoding="utf-8") as file_write:
        if file_name != "":
            data_file = get_data_file(file_prefix + file_name)
            results, total = count_words(data_file)
            print_results(results, total, file_write)
        end_time = time.time()
        string_time = "Execution Time " + str(end_time - start_time) + "s"
        print(string_time)
        file_write.write(string_time + "\n")
        file_write.close()
```

```
PS C:\Users\CHECH\OneDrive\Documentos\TEC\PruebasSoftware\EjerciciosPrácticos\A01794935_A4.2> pylint word_count.py

-------------------------------------------------------------------
Your code has been rated at 10.00/10 (previous run: 9.82/10, +0.18)
```

The code is executed successfully.



```python

def main():
    """Principal Main Compute Statistics Function."""
    start_time = time.time()
    file_prefix = "ArchivosApoyo/P3/"
    file_name = get_name_file()
    with open ("ConversionResults.txt", "w", encoding="utf-8") as file_write:
        if file_name != "":
            data_file = get_data_file(file_prefix + file_name)
            results, total = count_words(data_file)
            print_results(results, total, file_write)
        end_time = time.time()
        string_time = "Execution Time " + str(end_time - start_time) + "s"
        print(string_time)
        file_write.write(string_time + "\n")
        file_write.close()
```

```
PS C:\Users\CHECH\OneDrive\Documentos\TEC\PruebasSoftware\EjerciciosPrácticos\A01794935_A4.2> pylint word_count.py

-------------------------------------------------------------------
Your code has been rated at 10.00/10 (previous run: 9.82/10, +0.18)

PS C:\Users\CHECH\OneDrive\Documentos\TEC\PruebasSoftware\EjerciciosPrácticos\A01794935_A4.2> python word_count.py TC1.txt
conservative 2
mother 1
tions 1
pin 1
sure 1
regulatory 1
shower 1
uni 1
dial 1
photography 1
buying 1
```