



Tecnológico de Monterrey

ACTIVIDAD SEMANA 4

ACTIVIDAD 4.2: Ejercicio de programación 1

MATERIA: PRUEBAS DE SOFTWARE Y ASEGURAMIENTO DE LA CALIDAD

ALUMNO: DANIEL HERNANDEZ MORA

MATRICULA: A01793538

PROFESOR TITULAR: DR GERARDO PADILLA ZÁRATE

TUTOR: MTRA. YETNALEZI QUINTAS RUIZ

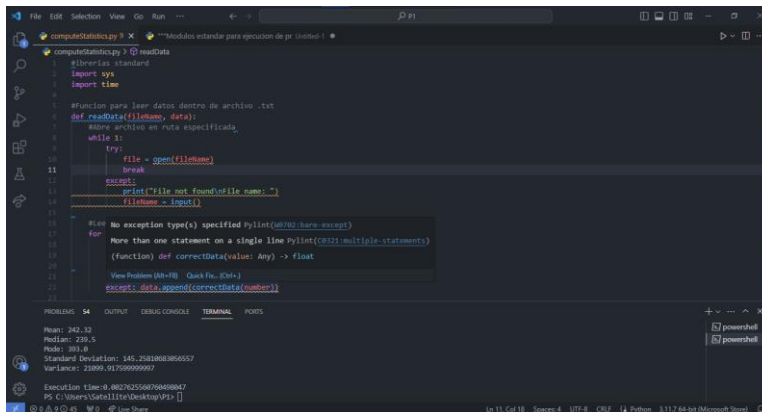
FECHA: 04/02/2024

CDMX

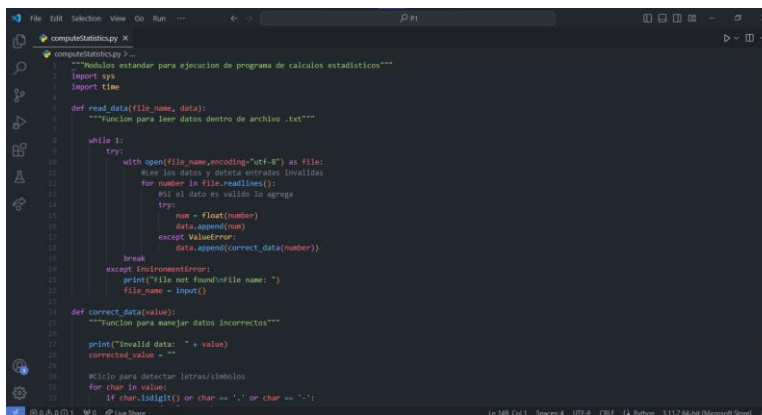


compute_statistics.py

PYLINT PRE

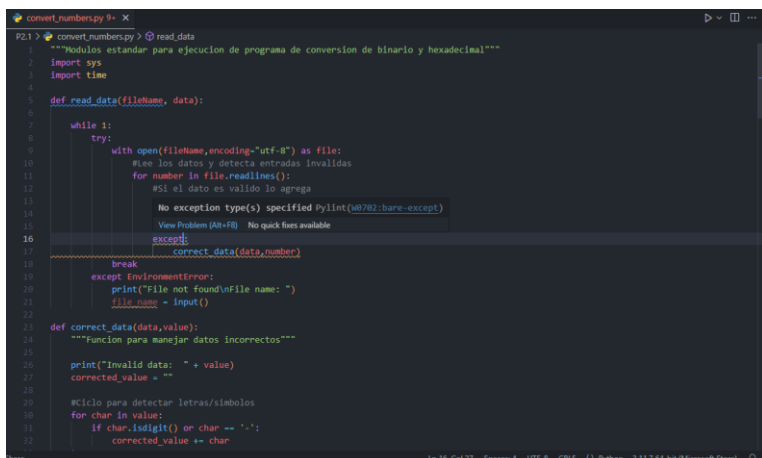


PYLINT POS



convert_numbers.py

PYLINT PRE



PYLINT POS

```
convert_numbers.py X
P3.1 > convert_numbers.py > read_data
1 """Modulos estandar para ejecucion de programa de conversion de binario y hexadecimal"""
2 import sys
3 import time
4
5 def read_data(file_name, data):
6     """Funcion para leer datos dentro de archivo .txt"""
7
8     while 1:
9         try:
10             with open(file_name, encoding="utf-8") as file:
11                 #Lee los datos y detecta entradas invalidas
12                 for number in file.readlines():
13                     #Si el dato es valido lo agrega
14                     try:
15                         num = int(number)
16                         data.append(num)
17                     except ValueError:
18                         correct_data(data, number)
19                 break
20             except EnvironmentError:
21                 print("File not found\nFile name: ")
22                 file_name = input()
23
24 def correct_data(data, value):
25     """Funcion para manejar datos incorrectos"""
26
27     print("Invalid data: " + value)
28     correctd_value = ""
29
30     #Ciclo para detectar letras/simbolos
31     for char in value:
32         if char.isdigit() or char == '.':
```

word_count.py

PYLINT PRE

```
WordCountResults.py 7
P3 > WordCountResults.py > final_time
1 """Modulos estandar para ejecucion de programa de conversion de binario y hexadecimal"""
2 import sys
3 import time
4
5 def read_data(file_name):
6     """Funcion para leer datos dentro de archivo .txt"""
7
8     while 1:
9         try:
10             with open(file_name, encoding="utf-8") as file:
11                 #Lee los datos y detecta entradas invalidas
12                 try:
13                     text = file.read()
14                     return clear_data(text)
15                 except:
16                     print("Error in data, try other file: ")
17                     file_name = input()
18                     continue
19             break
20             except EnvironmentError:
21                 print("File not found\nFile name: ")
22                 file_name = input()
23
24 def clear_data(text):
25     """Funcion para encontrar palabras repetidas"""
26
27     clean_data = []
28     for word in text.split('\n'):
29         clean_data.append(word)
30     return clean_data
31
32 def word_repeat(data):
```

PYLINT POS

```
word_count.py X
P3 > word_count.py > main
1 """Modulos estandar para ejecucion de programa de conversion de binario y hexadecimal"""
2 import sys
3 import time
4
5 def read_data(file_name):
6     """Funcion para leer datos dentro de archivo .txt"""
7
8     while 1:
9         try:
10             with open(file_name, encoding="utf-8") as file:
11                 #Lee los datos y detecta entradas invalidas
12                 try:
13                     text = file.read()
14                     return clear_data(text)
15                 except ValueError:
16                     print("Error in data, try other file: ")
17                     file_name = input()
18                     continue
19             break
20             except EnvironmentError:
21                 print("File not found\nFile name: ")
22                 file_name = input()
23
24 def clear_data(text):
25     """Funcion para encontrar palabras repetidas"""
26
27     clean_data = []
28     for word in text.split('\n'):
29         clean_data.append(word)
30     return clean_data
31
32 def word_repeat(data):
```

P1 Outputs

```
File Edit Selection View Go Run
compute_statistics.py X
OPEN EDITORS
compute_statistics.py P1.2
def read_data(file_name, data):
    """Funcion para leer datos dentro de archivo .txt"""
    while 1:
        try:
            with open(file_name, encoding="utf-8") as file:
                # Lee los datos y detecta entradas invalidas
                for number in file.readlines():
                    # Si el dato es valido lo agrega
                    try:
                        num = float(number)
                    except:
                        continue
        except:
            continue
    return data

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Satellite\Desktop\vm\gl.2> python compute_statistics.py TC3.txt
Total: 3000
Mean: 1.0798869276747e+20
Median: 1.08888888888888e+20
Mode: N/A EXISTE MAS DE UNA MODA
Standard Deviation: 1.0798869276747e+20
Variance: 1.15309886953804e+40
Execution time: 8.1544862780888496
PS C:\Users\Satellite\Desktop\vm\gl.2>
```

```
File Edit Selection View Go Run
compute_statistics.py X
OPEN EDITORS
compute_statistics.py P1.2
def read_data(file_name, data):
    """Funcion para leer datos dentro de archivo .txt"""
    while 1:
        try:
            with open(file_name, encoding="utf-8") as file:
                # Lee los datos y detecta entradas invalidas
                for number in file.readlines():
                    # Si el dato es valido lo agrega
                    try:
                        num = float(number)
                    except:
                        continue
        except:
            continue
    return data

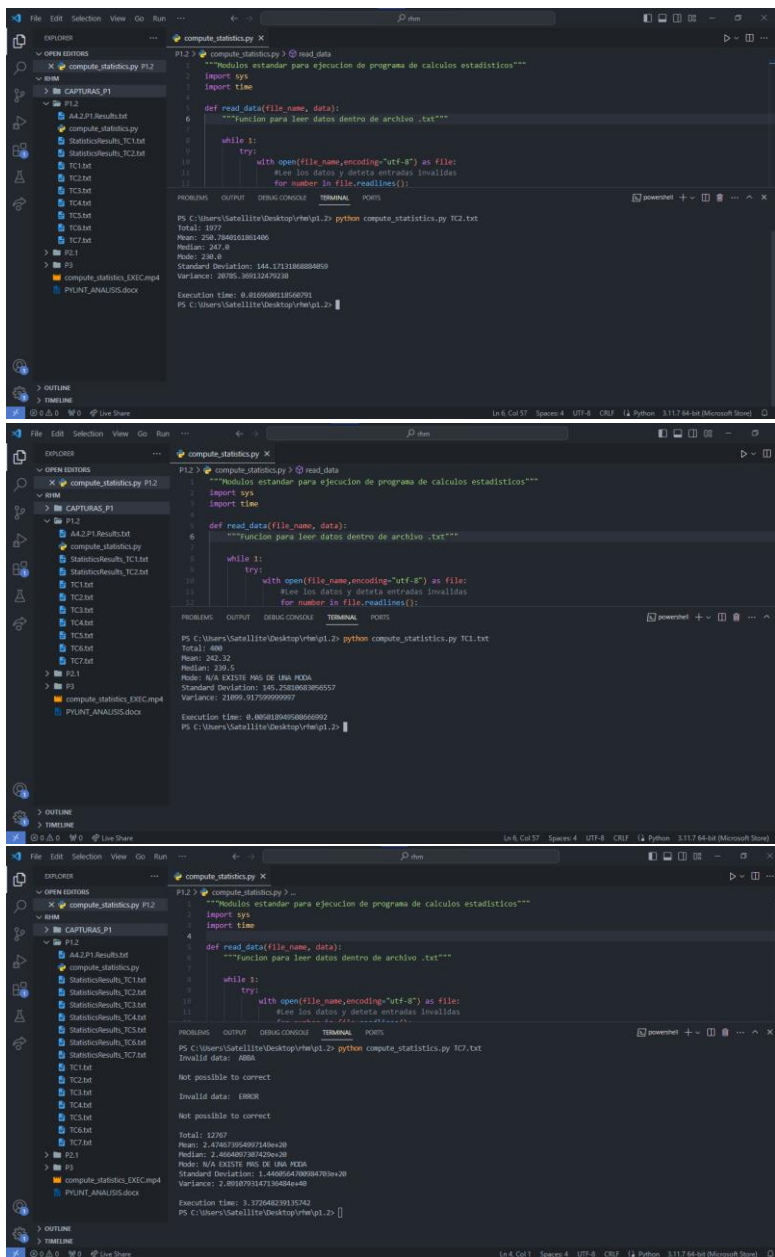
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Satellite\Desktop\vm\gl.2> python compute_statistics.py TC3.txt
Invalid data: AAA
Not possible to correct
Invalid data: 23,45
Corrected: 23.45
Invalid data: 11;54
Corrected: 11.54
Invalid data: 11
Not possible to correct
Total: 309
Mean: 240.49527588888612
Median: 241.4
Mode: N/A EXISTE MAS DE UNA MODA
Standard Deviation: 240.49527588888612
Variance: 2145.9554713428
Execution time: 8.0883222525044486
PS C:\Users\Satellite\Desktop\vm\gl.2>
```

```
File Edit Selection View Go Run
compute_statistics.py X
OPEN EDITORS
compute_statistics.py P1.2
def read_data(file_name, data):
    """Funcion para leer datos dentro de archivo .txt"""
    while 1:
        try:
            with open(file_name, encoding="utf-8") as file:
                # Lee los datos y detecta entradas invalidas
                for number in file.readlines():
                    # Si el dato es valido lo agrega
                    try:
                        num = float(number)
                    except:
                        continue
        except:
            continue
    return data

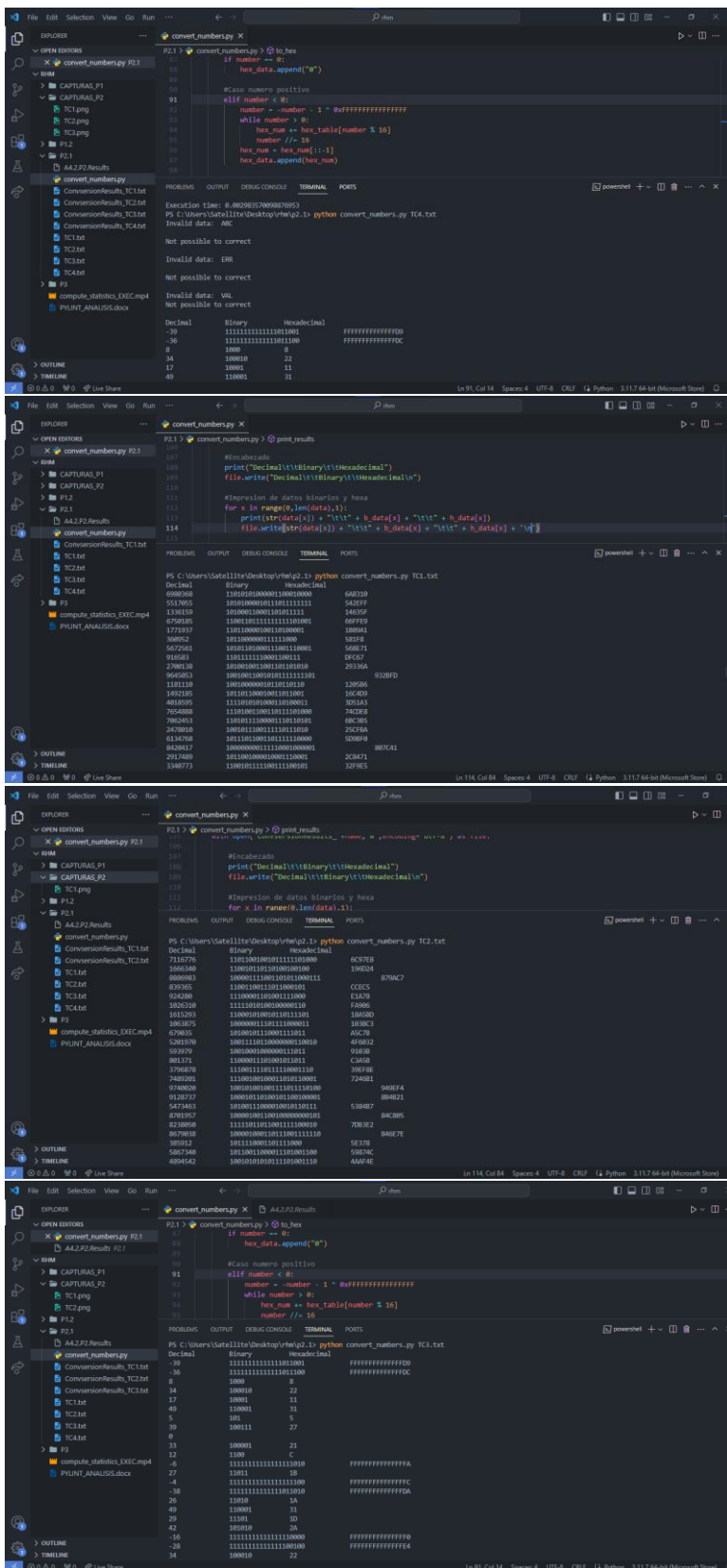
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Satellite\Desktop\vm\gl.2> python compute_statistics.py TC4.txt
Total: 12024
Mean: 149.88267347682746
Median: 147.75
Mode: 123.25
Standard Deviation: 138.43443961368894
Variance: 17807.328843018837
Execution time: 8.1850855682676227
PS C:\Users\Satellite\Desktop\vm\gl.2>
```

```
File Edit Selection View Go Run
compute_statistics.py X
OPEN EDITORS
compute_statistics.py P1.2
def read_data(file_name, data):
    """Funcion para leer datos dentro de archivo .txt"""
    while 1:
        try:
            with open(file_name, encoding="utf-8") as file:
                # Lee los datos y detecta entradas invalidas
                for number in file.readlines():
                    # Si el dato es valido lo agrega
                    try:
                        num = float(number)
                    except:
                        continue
        except:
            continue
    return data

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Satellite\Desktop\vm\gl.2> python compute_statistics.py TC1.txt
Total: 12024
Mean: 240.77621888888583
Median: 240.8
Mode: 94.0
Standard Deviation: 145.31784888817962
Variance: 21117.2774733429
Execution time: 8.12574338812963827
PS C:\Users\Satellite\Desktop\vm\gl.2>
```



P2 Outputs



P3 Outputs

