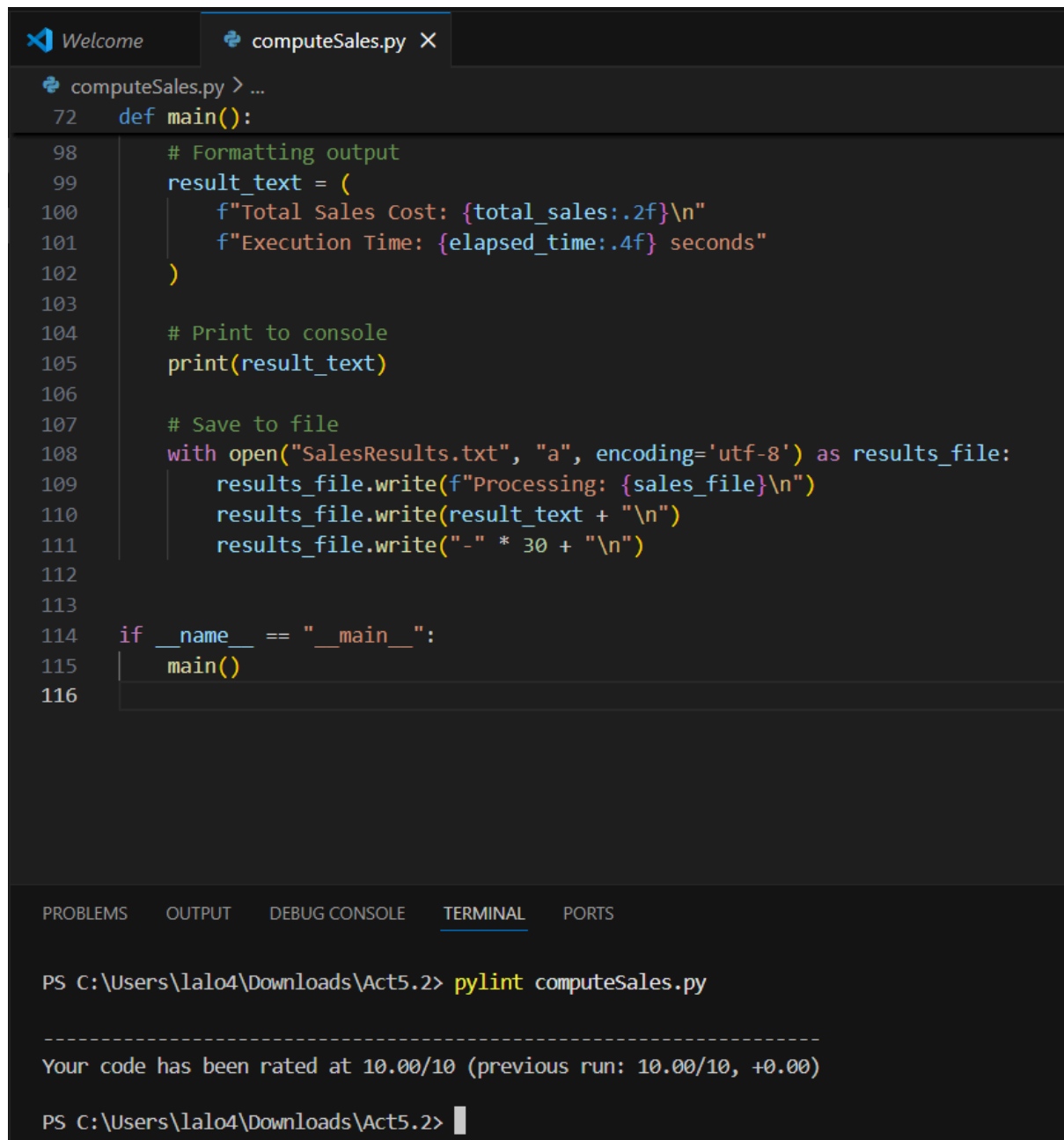


## Evidencia Pylint



The image shows a code editor with a dark theme. The top bar has a 'Welcome' tab and a 'computeSales.py' tab. The editor displays the following Python code:

```
72 def main():
98     # Formatting output
99     result_text = (
100         f"Total Sales Cost: {total_sales:.2f}\n"
101         f"Execution Time: {elapsed_time:.4f} seconds"
102     )
103
104     # Print to console
105     print(result_text)
106
107     # Save to file
108     with open("SalesResults.txt", "a", encoding='utf-8') as results_file:
109         results_file.write(f"Processing: {sales_file}\n")
110         results_file.write(result_text + "\n")
111         results_file.write("-" * 30 + "\n")
112
113
114 if __name__ == "__main__":
115     main()
116
```

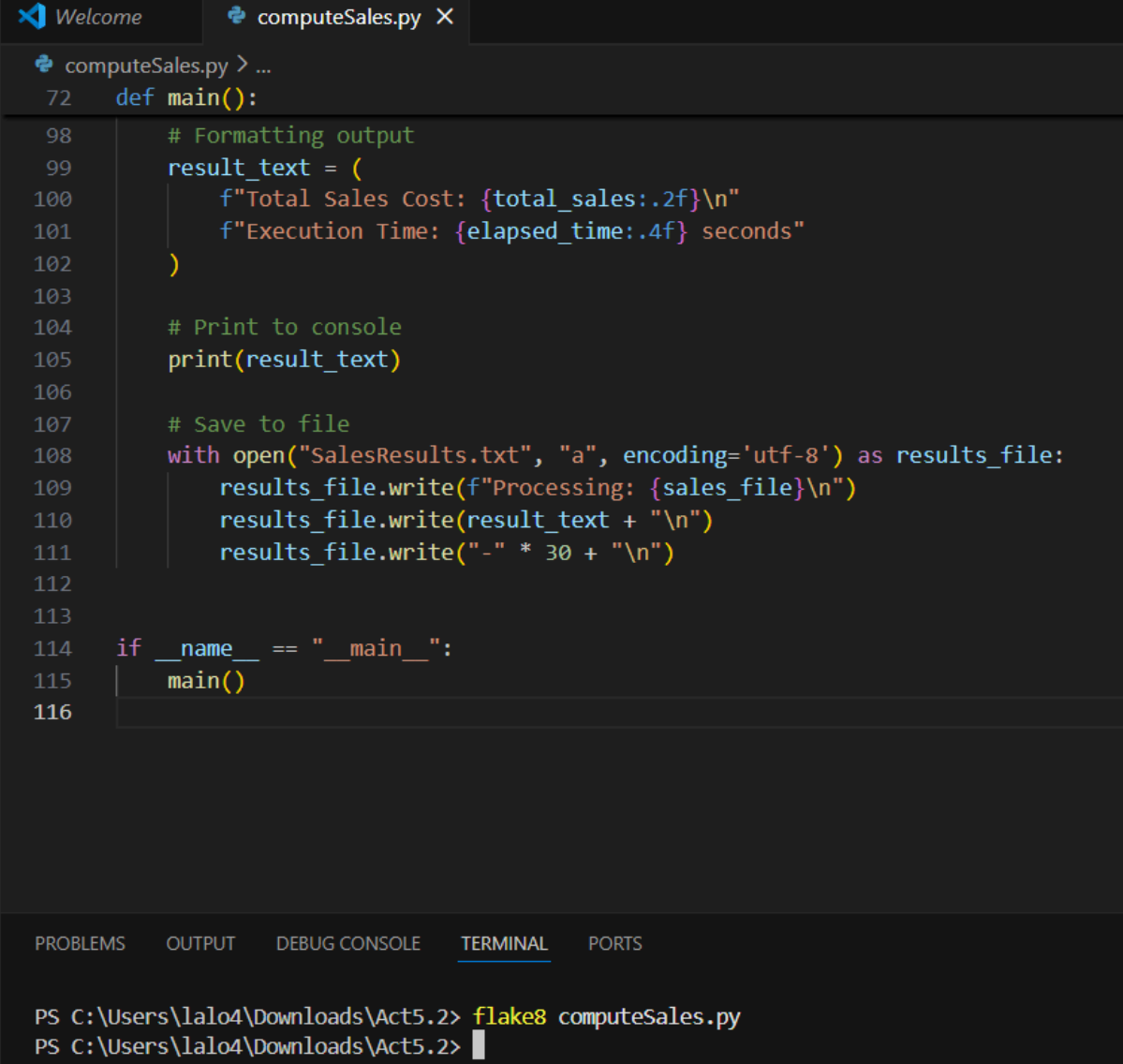
Below the code editor is a terminal window with tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', and 'PORTS'. The 'TERMINAL' tab is active, showing the following text:

```
PS C:\Users\lalo4\Downloads\Act5.2> pylint computeSales.py

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

PS C:\Users\lalo4\Downloads\Act5.2> 
```

## Evidencia Flake8



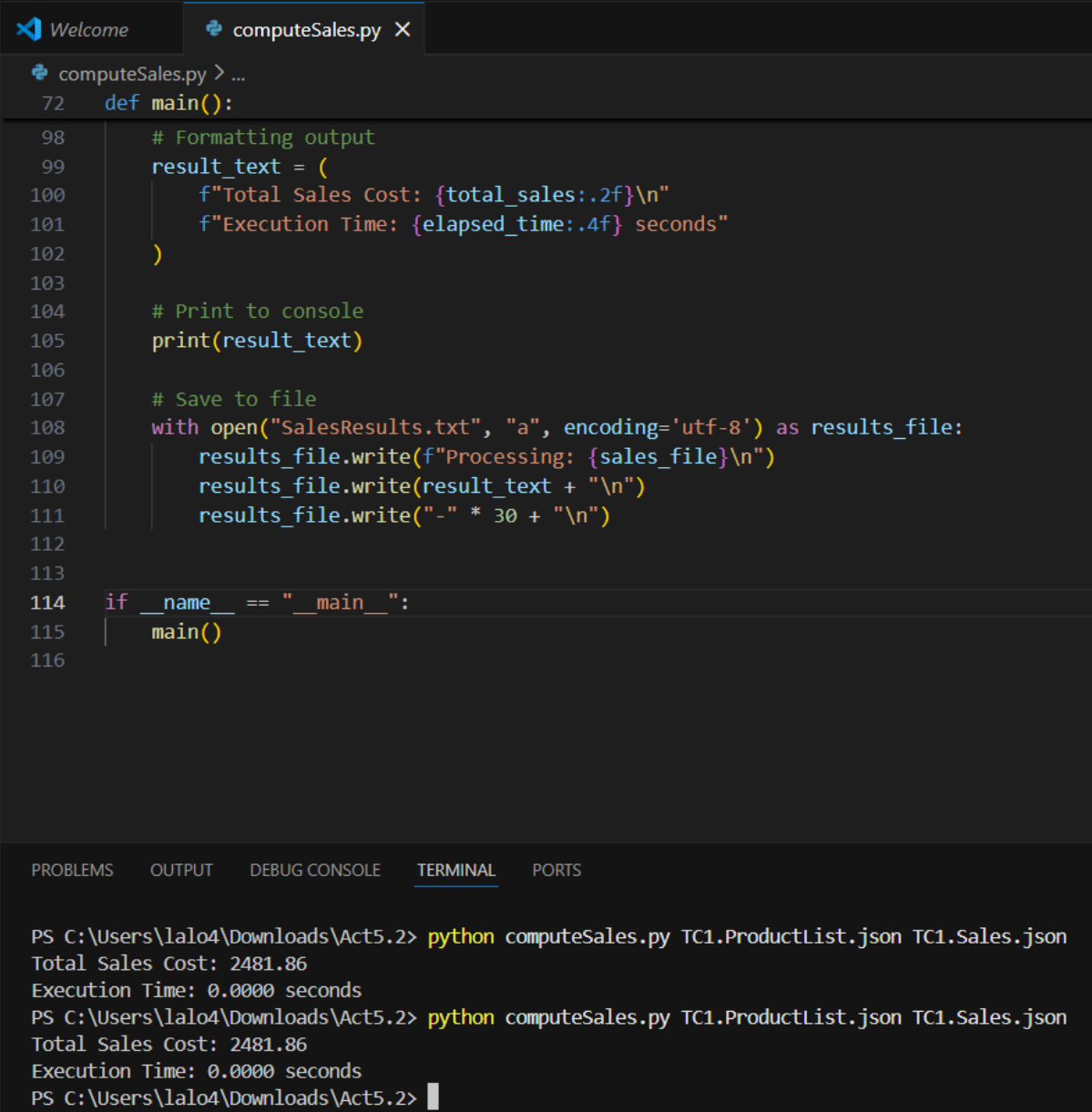
The image shows a code editor with a dark theme. At the top, there are two tabs: 'Welcome' and 'computeSales.py'. The 'computeSales.py' tab is active, showing a Python script. The script starts with a `def main():` function. Inside this function, there are several lines of code: a comment `# Formatting output`, a variable `result_text` assigned a multi-line string with formatted values for `total_sales` and `elapsed_time`, a comment `# Print to console`, a `print(result_text)` statement, a comment `# Save to file`, and a `with open` block that writes 'Processing: {sales\_file}' followed by the `result_text` and a separator line of 30 dashes. After the function, there is an `if __name__ == '__main__':` block that calls `main()`. The line numbers 72, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, and 116 are visible on the left side of the editor. At the bottom of the editor, there is a terminal window with tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', and 'PORTS'. The 'TERMINAL' tab is active, showing the command `flake8 computeSales.py` being executed in a PowerShell prompt at the path `C:\Users\lalo4\Downloads\Act5.2`.

```
72  def main():
98      # Formatting output
99      result_text = (
100          f"Total Sales Cost: {total_sales:.2f}\n"
101          f"Execution Time: {elapsed_time:.4f} seconds"
102      )
103
104      # Print to console
105      print(result_text)
106
107      # Save to file
108      with open("SalesResults.txt", "a", encoding='utf-8') as results_file:
109          results_file.write(f"Processing: {sales_file}\n")
110          results_file.write(result_text + "\n")
111          results_file.write("-" * 30 + "\n")
112
113
114  if __name__ == "__main__":
115      main()
116
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\lalo4\Downloads\Act5.2> flake8 computeSales.py  
PS C:\Users\lalo4\Downloads\Act5.2>

## Evidencia TC1



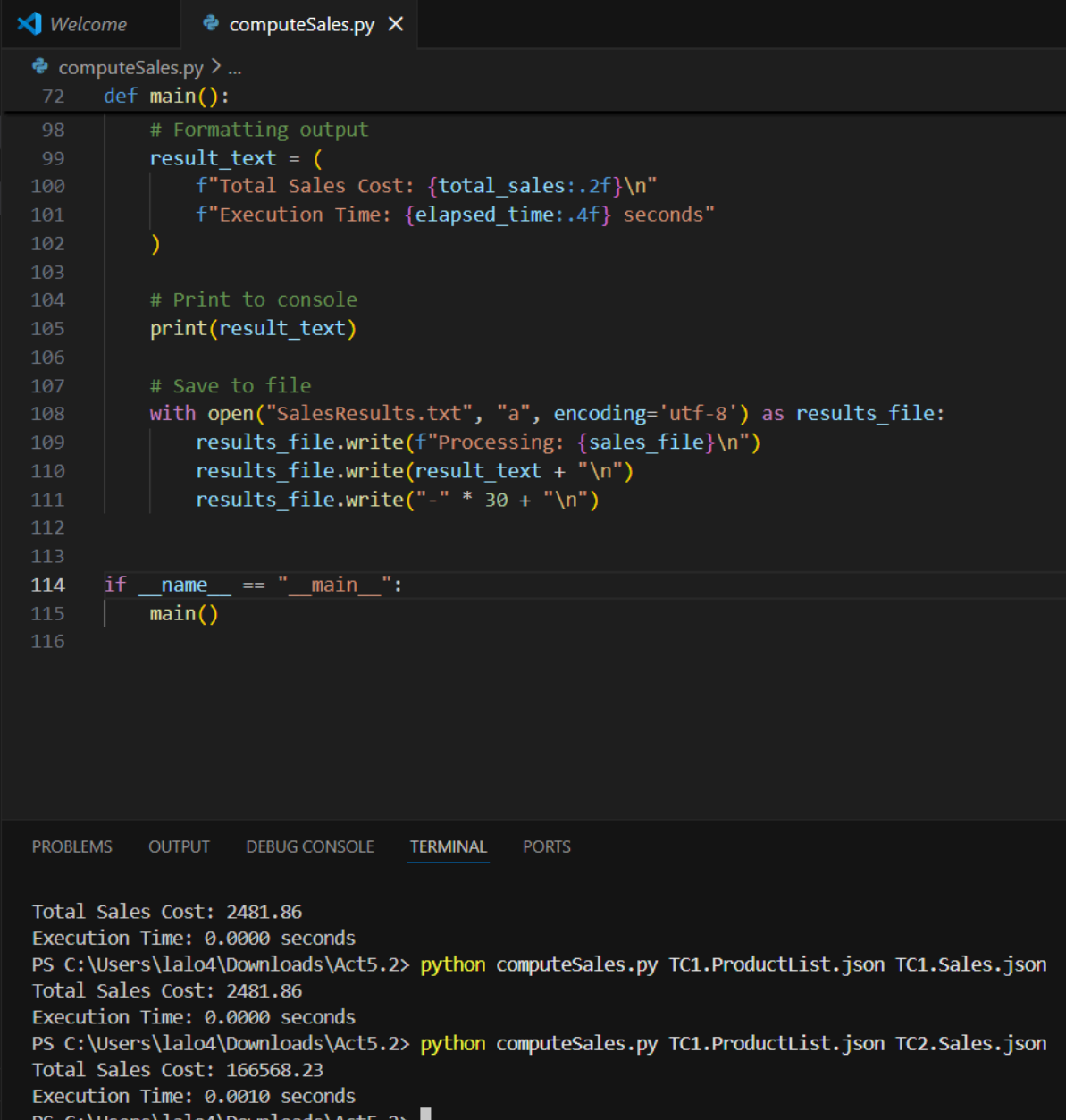
The image shows a Visual Studio Code editor window with a file named `computeSales.py` open. The script defines a `main()` function that formats output, prints it to the console, and saves it to a file named `SalesResults.txt`. The terminal at the bottom shows the script being executed twice, both times producing the same output: `Total Sales Cost: 2481.86` and `Execution Time: 0.0000 seconds`.

```
72 def main():
98     # Formatting output
99     result_text = (
100         f"Total Sales Cost: {total_sales:.2f}\n"
101         f"Execution Time: {elapsed_time:.4f} seconds"
102     )
103
104     # Print to console
105     print(result_text)
106
107     # Save to file
108     with open("SalesResults.txt", "a", encoding='utf-8') as results_file:
109         results_file.write(f"Processing: {sales_file}\n")
110         results_file.write(result_text + "\n")
111         results_file.write("-" * 30 + "\n")
112
113
114 if __name__ == "__main__":
115     main()
116
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\lalo4\Downloads\Act5.2> python computeSales.py TC1.ProductList.json TC1.Sales.json
Total Sales Cost: 2481.86
Execution Time: 0.0000 seconds
PS C:\Users\lalo4\Downloads\Act5.2> python computeSales.py TC1.ProductList.json TC1.Sales.json
Total Sales Cost: 2481.86
Execution Time: 0.0000 seconds
PS C:\Users\lalo4\Downloads\Act5.2>
```

## Evidencia TC2



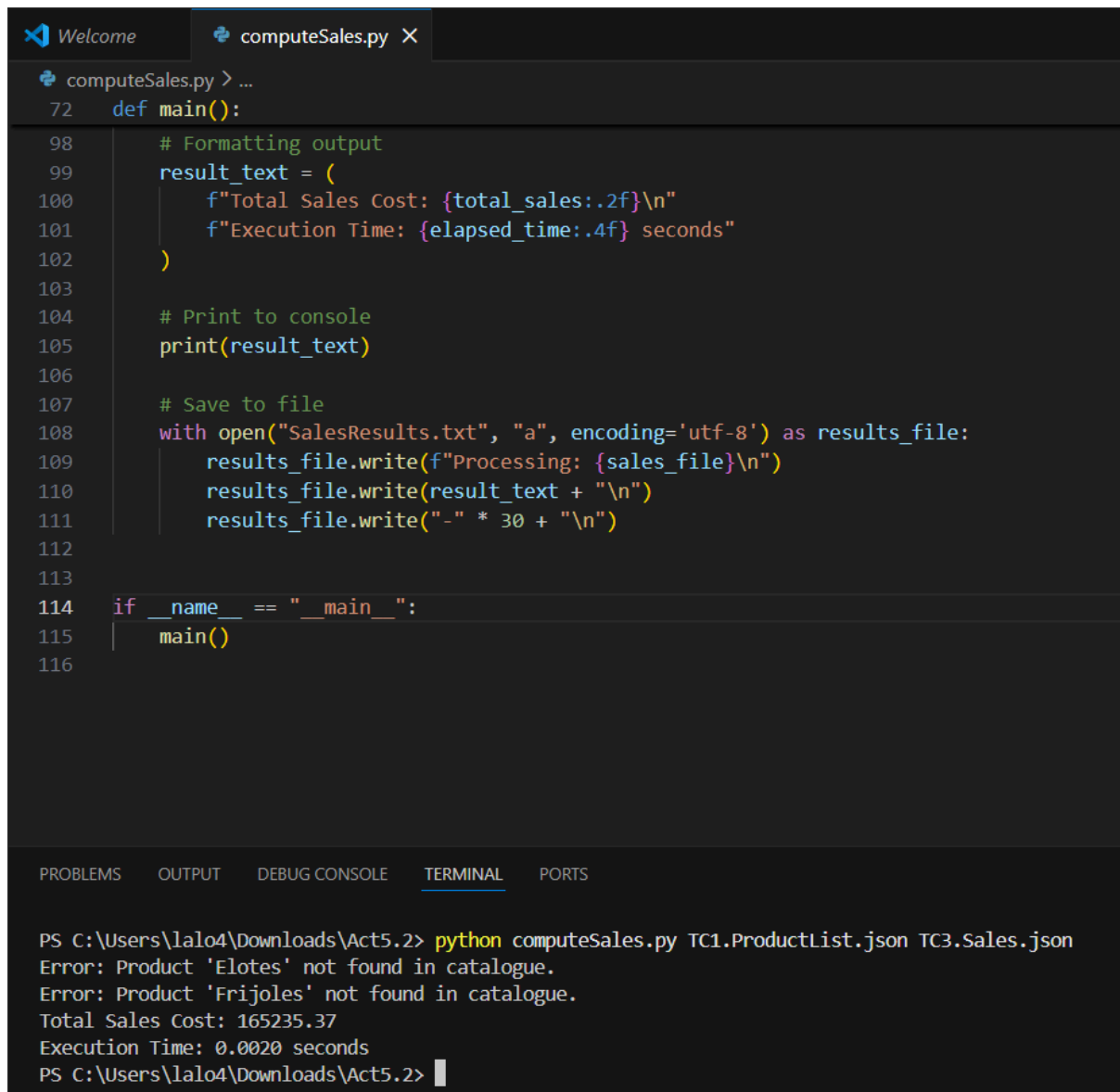
The image shows a Visual Studio Code editor window with a file named `computeSales.py`. The script defines a `main()` function that formats and prints sales data, then saves it to a file. The terminal at the bottom shows the script being executed twice with different input files, producing the expected output.

```
computeSales.py > ...
72 def main():
73     # Formatting output
74     result_text = (
75         f"Total Sales Cost: {total_sales:.2f}\n"
76         f"Execution Time: {elapsed_time:.4f} seconds"
77     )
78
79     # Print to console
80     print(result_text)
81
82     # Save to file
83     with open("SalesResults.txt", "a", encoding='utf-8') as results_file:
84         results_file.write(f"Processing: {sales_file}\n")
85         results_file.write(result_text + "\n")
86         results_file.write("-" * 30 + "\n")
87
88 if __name__ == "__main__":
89     main()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Total Sales Cost: 2481.86  
Execution Time: 0.0000 seconds  
PS C:\Users\lalo4\Downloads\Act5.2> python computeSales.py TC1.ProductList.json TC1.Sales.json  
Total Sales Cost: 2481.86  
Execution Time: 0.0000 seconds  
PS C:\Users\lalo4\Downloads\Act5.2> python computeSales.py TC1.ProductList.json TC2.Sales.json  
Total Sales Cost: 166568.23  
Execution Time: 0.0010 seconds  
PS C:\Users\lalo4\Downloads\Act5.2>

## Evidencia TC3



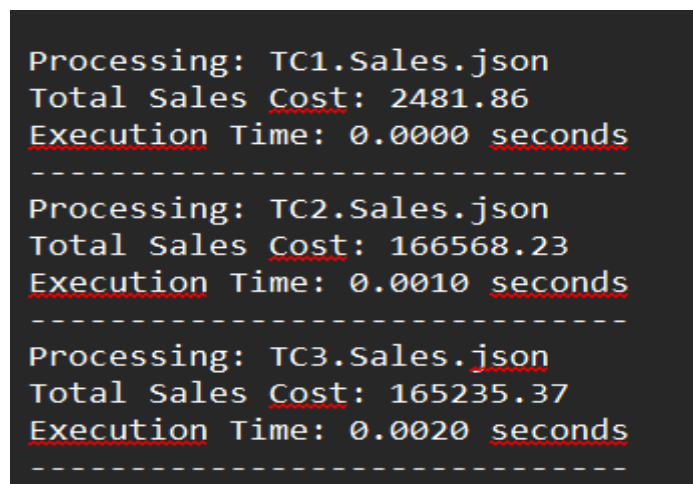
The image shows a Visual Studio Code editor window with a file named `computeSales.py`. The script defines a `main()` function that formats output, prints it to the console, and saves it to a file named `SalesResults.txt`. The terminal shows the execution of the script, which processes `TC1.ProductList.json` and `TC3.Sales.json`, resulting in errors for missing products and specific sales data.

```
72 def main():
98     # Formatting output
99     result_text = (
100         f"Total Sales Cost: {total_sales:.2f}\n"
101         f"Execution Time: {elapsed_time:.4f} seconds"
102     )
103
104     # Print to console
105     print(result_text)
106
107     # Save to file
108     with open("SalesResults.txt", "a", encoding='utf-8') as results_file:
109         results_file.write(f"Processing: {sales_file}\n")
110         results_file.write(result_text + "\n")
111         results_file.write("-" * 30 + "\n")
112
113
114 if __name__ == "__main__":
115     main()
116
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\lalo4\Downloads\Act5.2> python computeSales.py TC1.ProductList.json TC3.Sales.json
Error: Product 'Elotes' not found in catalogue.
Error: Product 'Frijoles' not found in catalogue.
Total Sales Cost: 165235.37
Execution Time: 0.0020 seconds
PS C:\Users\lalo4\Downloads\Act5.2>
```

## Resultados



The image shows the output of the script, which displays the processing status for three different sales files. Each entry shows the file name, the total sales cost, and the execution time, separated by a dashed line.

```
Processing: TC1.Sales.json
Total Sales Cost: 2481.86
Execution Time: 0.0000 seconds
-----
Processing: TC2.Sales.json
Total Sales Cost: 166568.23
Execution Time: 0.0010 seconds
-----
Processing: TC3.Sales.json
Total Sales Cost: 165235.37
Execution Time: 0.0020 seconds
-----
```