

GIRUGULA VARSHINI

2403A51L14

Batch : 51

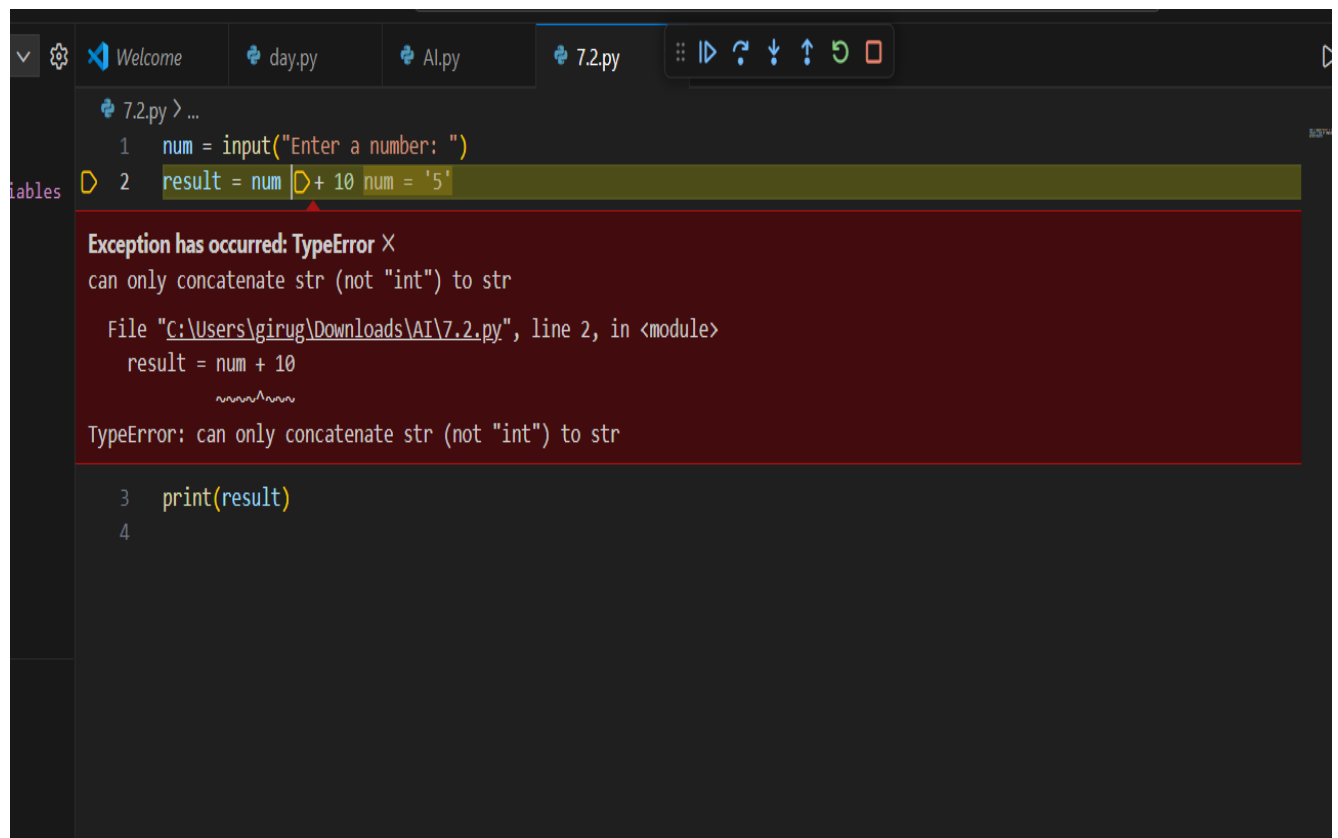
Lab 7: Error Debugging with AI: Systematic approaches to finding and fixing bugs

Task 1 – Runtime Error Due to Invalid Input Type

Prompt:

Write a Python function to determine whether a given number is prime.

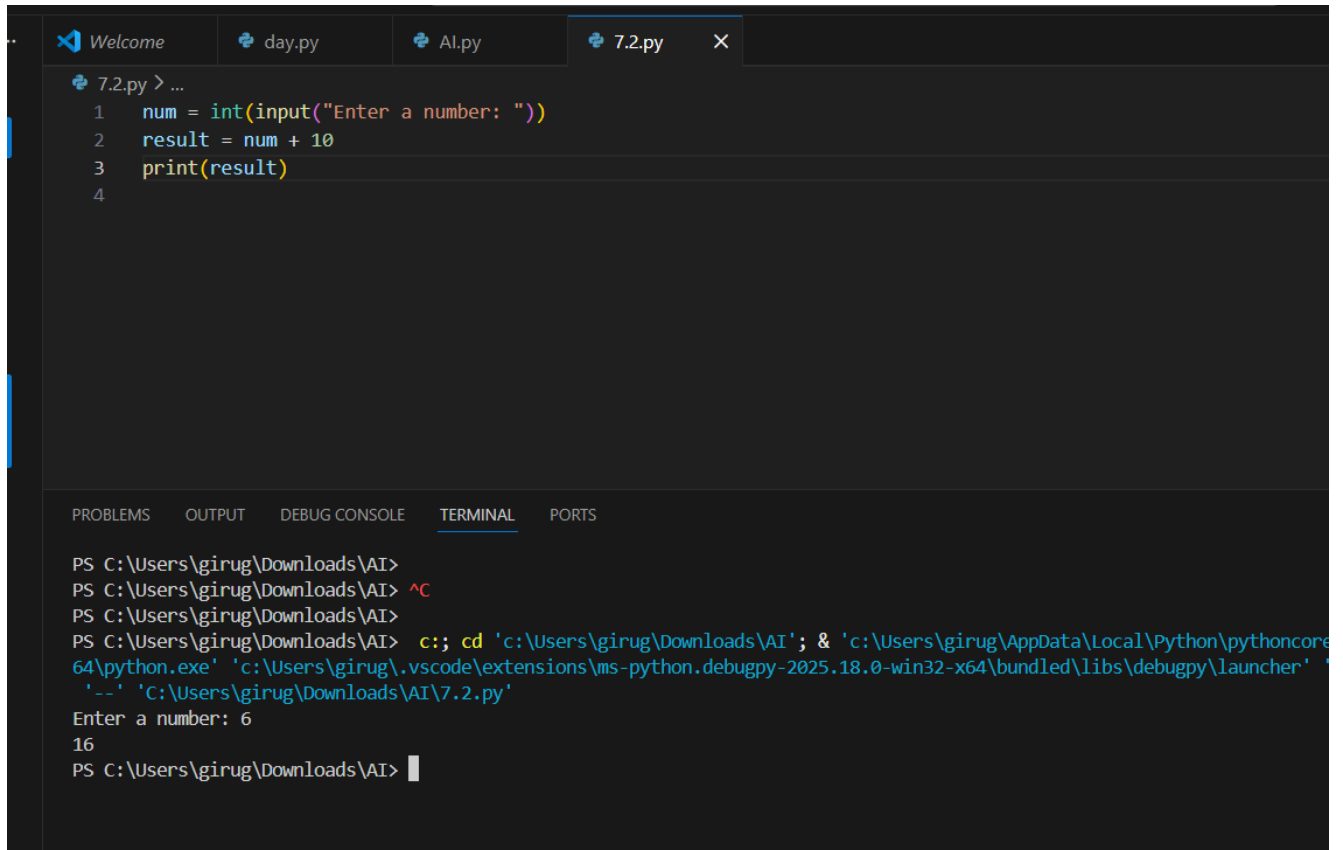
Code :



```
7.2.py > ...
1 num = input("Enter a number: ")
2 result = num + 10 num = '5'
3 print(result)
4
```

Exception has occurred: TypeError X
can only concatenate str (not "int") to str
File "C:\Users\girug\Downloads\AI\7.2.py", line 2, in <module>
result = num + 10
~~~~~  
TypeError: can only concatenate str (not "int") to str

## Output:



The screenshot shows the Visual Studio Code interface. At the top, there are four tabs: 'Welcome', 'day.py', 'AI.py', and '7.2.py'. The '7.2.py' tab is active. Below the tabs, the editor displays the following Python code:

```
1 num = int(input("Enter a number: "))
2 result = num + 10
3 print(result)
4
```

At the bottom of the window, the 'TERMINAL' panel is open. It shows the command prompt output for running the script:

```
PS C:\Users\girug\Downloads\AI>
PS C:\Users\girug\Downloads\AI> ^C
PS C:\Users\girug\Downloads\AI>
PS C:\Users\girug\Downloads\AI> c:: cd 'c:\Users\girug\Downloads\AI'; & 'c:\Users\girug\AppData\Local\Python\pythoncore
64\python.exe' 'c:\Users\girug\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '
' -- 'C:\Users\girug\Downloads\AI\7.2.py'
Enter a number: 6
16
PS C:\Users\girug\Downloads\AI>
```

**Justification:** The program failed because `input()` returns data as a string, and adding a string to an integer is invalid in Python. The AI corrected this by converting the input to an integer using `int()`, ensuring the arithmetic operation works properly. This type conversion is necessary to match the expected numeric behavior of the program.

## Task 2 – Incorrect Function Return Value

### Prompt:

Generate a function to calculate the sum of elements in a list.

## Code :

```
23
24
25
26
27
28 def square(n):
29     result = n * n
30
31
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\girug\Downloads\AI> c::; cd 'c:\Users\girug\Downloads\AI'; & 'c:\Users\girug\AppData\Local\Python\pythoncore-3.14-64\python.exe' 'c:\Users\girug\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '51096'
'--' 'C:\Users\girug\Downloads\AI\7.2.py'
PS C:\Users\girug\Downloads\AI> 6
6
PS C:\Users\girug\Downloads\AI> 
```

## Output:

```
24
25
26
27 def square(n):
28     result = n * n
29     return result
30 print(square(8))
31
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
> c::; cd 'c:\Users\girug\Downloads\AI'; & 'c:\Users\girug\AppData\Local\Python\pythoncore-3.14-64\python.exe' 'c:\Users\girug\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '638'
'--' 'C:\Users\girug\Downloads\AI\7.2.py'
25
PS C:\Users\girug\Downloads\AI> ^C
PS C:\Users\girug\Downloads\AI>
PS C:\Users\girug\Downloads\AI> c::; cd 'c:\Users\girug\Downloads\AI'; & 'c:\Users\girug\AppData\Local\Python\pythoncore-3.14-64\python.exe' 'c:\Users\girug\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '639'
'--' 'C:\Users\girug\Downloads\AI\7.2.py'
PS C:\Users\girug\Downloads\AI> 8^C
PS C:\Users\girug\Downloads\AI>
PS C:\Users\girug\Downloads\AI> c::; cd 'c:\Users\girug\Downloads\AI'; & 'c:\Users\girug\AppData\Local\Python\pythoncore-3.14-64\python.exe' 'c:\Users\girug\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '502'
'--' 'C:\Users\girug\Downloads\AI\7.2.py'
64
PS C:\Users\girug\Downloads\AI> 
```

Ln 30, Col 15

## Justification:

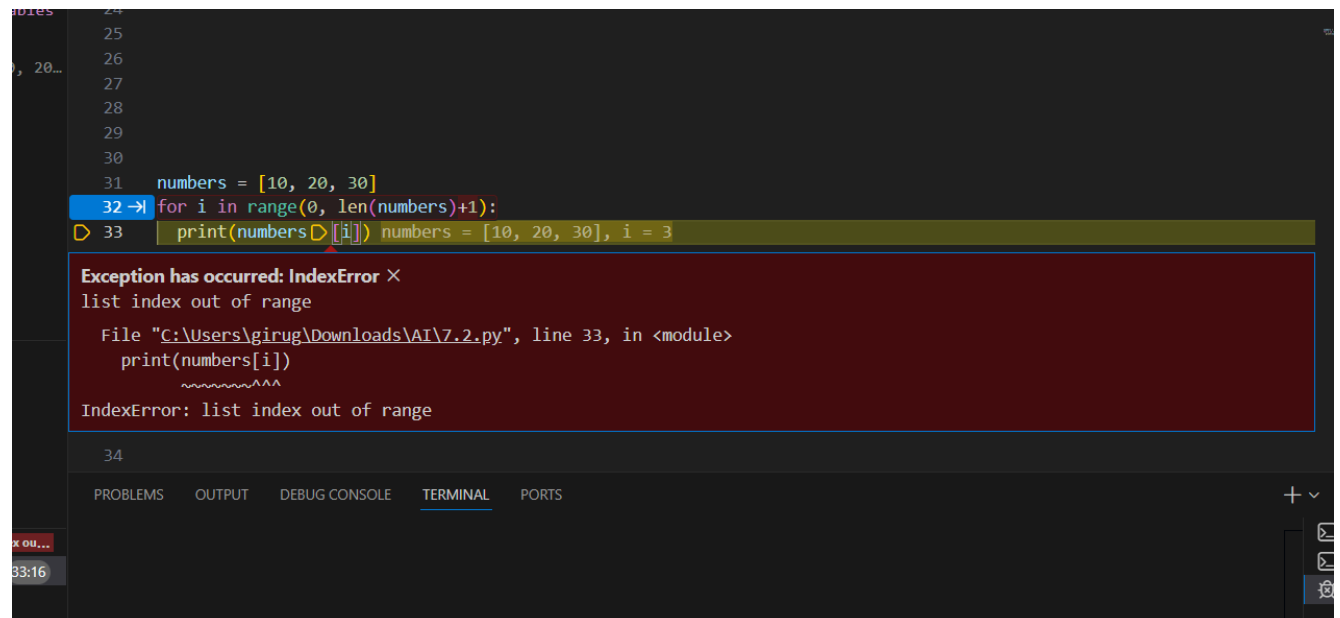
Although the function computed the square internally, it never returned the result, causing the output to be lost. The AI identified the missing return statement and added it, allowing the function to properly

send the computed value back to the caller. Returning values is essential for functional correctness and reusability.

### Task 3 – IndexError in List Traversal

Write a Python function that takes an alphanumeric string and returns only the digits.

**Code :**



```
24
25
26
27
28
29
30
31 numbers = [10, 20, 30]
32 → for i in range(0, len(numbers)+1):
33     print(numbers[i]) numbers = [10, 20, 30], i = 3
```

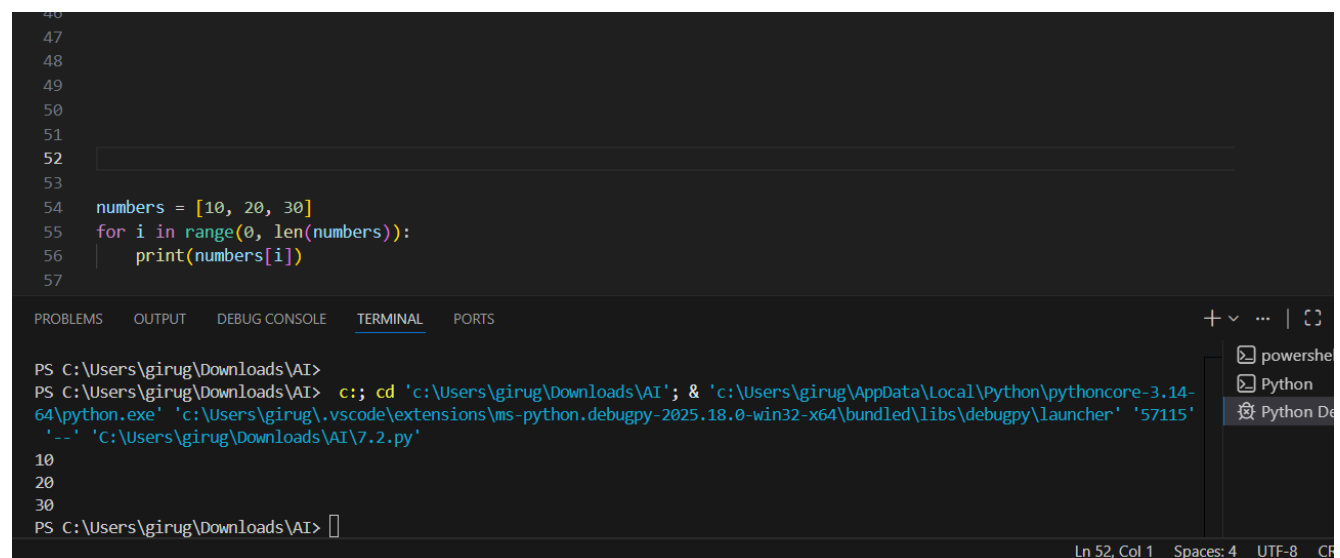
Exception has occurred: IndexError ×  
list index out of range

File "C:\Users\girug\Downloads\AI\7.2.py", line 33, in <module>  
 print(numbers[i])  
 ~~~~~^~^~  
IndexError: list index out of range

34

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Output:



```
40
47
48
49
50
51
52
53
54 numbers = [10, 20, 30]
55 for i in range(0, len(numbers)):
56     print(numbers[i])
57
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\girug\Downloads\AI>
PS C:\Users\girug\Downloads\AI> c:: cd 'c:\Users\girug\Downloads\AI'; & 'c:\Users\girug\AppData\Local\Python\pythoncore-3.14-64\python.exe' 'c:\Users\girug\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '57115' '--' 'C:\Users\girug\Downloads\AI\7.2.py'
10
20
30
PS C:\Users\girug\Downloads\AI>

Ln 52, Col 1 Spaces: 4 UTF-8 CP

Justification:

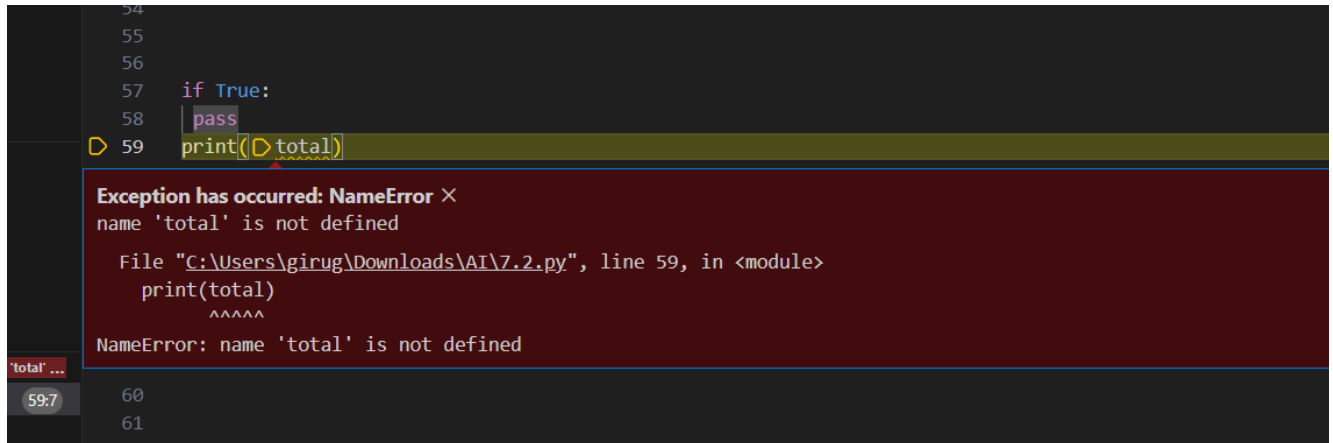
The loop incorrectly iterated one step beyond the valid index range using `len(numbers) + 1`, causing an `IndexError`. AI fixed the boundary to `range(len(numbers))`, ensuring safe access of all existing list elements. This correction is justified because valid indices only go from 0 to `len(numbers)-1`.

Task 4 – Uninitialized Variable Usage

Prompt :

Write a Python function to count the number of vowels in a given string.

Code :



The screenshot shows a Python IDE with a dark theme. The editor displays a Python script with the following code:

```
54
55
56
57 if True:
58     pass
59 print(total)
```

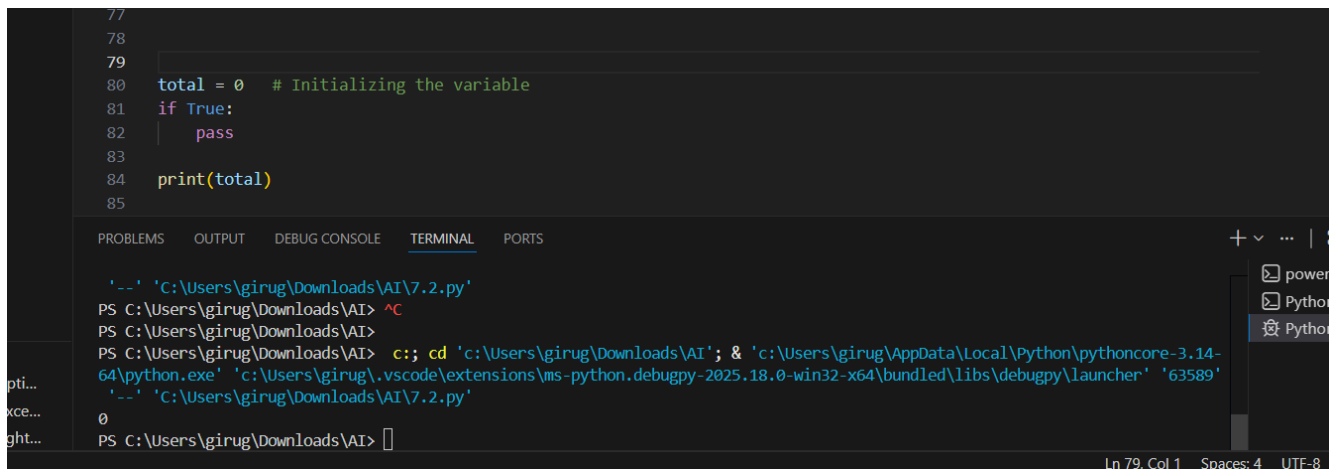
Line 59 is highlighted. A red error message box is overlaid on the code, stating:

```
Exception has occurred: NameError ×
name 'total' is not defined

File "C:\Users\girug\Downloads\AI\7.2.py", line 59, in <module>
    print(total)
    ^^^^^
NameError: name 'total' is not defined
```

The error message indicates that the variable 'total' has not been defined before it is used in the print statement.

Output:



The screenshot shows the same Python IDE with the corrected code:

```
77
78
79 total = 0 # Initializing the variable
80 if True:
81     pass
82
83 print(total)
84
85
```

The IDE's output window is open, showing the execution of the script:

```
-- 'C:\Users\girug\Downloads\AI\7.2.py'
PS C:\Users\girug\Downloads\AI> ^C
PS C:\Users\girug\Downloads\AI>
PS C:\Users\girug\Downloads\AI> c:: cd 'c:\Users\girug\Downloads\AI'; & 'c:\Users\girug\AppData\Local\Python\pythoncore-3.14-64\python.exe' 'c:\Users\girug\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundle\libs\debugpy\launcher' '63589'
-- 'C:\Users\girug\Downloads\AI\7.2.py'
0
PS C:\Users\girug\Downloads\AI>
```

The output shows that the variable 'total' is now initialized to 0, and the program runs successfully without any errors.

Justification:

The program attempted to print a variable (total) before it had been assigned any value, resulting in a runtime error. AI resolved this by initializing the variable to 0 before use, ensuring the program has a valid reference. Proper initialization prevents undefined behavior and is a fundamental programming requirement.

Task 5 – Logical Error in Student Grading System

Prompt :

write a Python function that takes three numbers and returns the minimum value without using min().

Code :

```
79
80 marks = 85
81 if marks >= 90:
82     grade = "A"
83 elif marks >= 80:
84     grade = "C"
85 else:
86     grade = "B"
87 print(grade)
88
89
90
91
92
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\girug\Downloads\AI> ^C
PS C:\Users\girug\Downloads\AI>
PS C:\Users\girug\Downloads\AI> c:: cd 'c:\Users\girug\Downloads\AI'; & 'c:\Users\girug\AppData\Local\Python\pythoncore-3.14-64\python.exe' 'c:\Users\girug\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '50305'
'--' 'C:\Users\girug\Downloads\AI\7.2.py'
PS C:\Users\girug\Downloads\AI> 
```

Output:

```
74
75
76
77
78
79 marks = 85
80
81 if marks >= 90:
82     grade = "A"
83 elif marks >= 80:
84     grade = "B"
85 else:
86     grade = "C"
87
88 print(grade)
89
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
C
PS C:\Users\girug\Downloads\AI> ^C
PS C:\Users\girug\Downloads\AI>
PS C:\Users\girug\Downloads\AI> c:: cd 'c:\Users\girug\Downloads\AI'; & 'c:\Users\girug\AppData\Local\Python\pythoncore-3.14-64\python.exe' 'c:\Users\girug\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '64291'
'--' 'C:\Users\girug\Downloads\AI\7.2.py'
B
PS C:\Users\girug\Downloads\AI> 
```

Justification:

The conditions for assigning grades were incorrectly ordered, making the program assign a wrong grade for certain mark ranges. AI fixed this by arranging the conditions in a logically descending order (A → B → C), ensuring accurate evaluation. Correct conditional structure is essential for producing correct program decisions.