

ASSIGNMENT-4.2

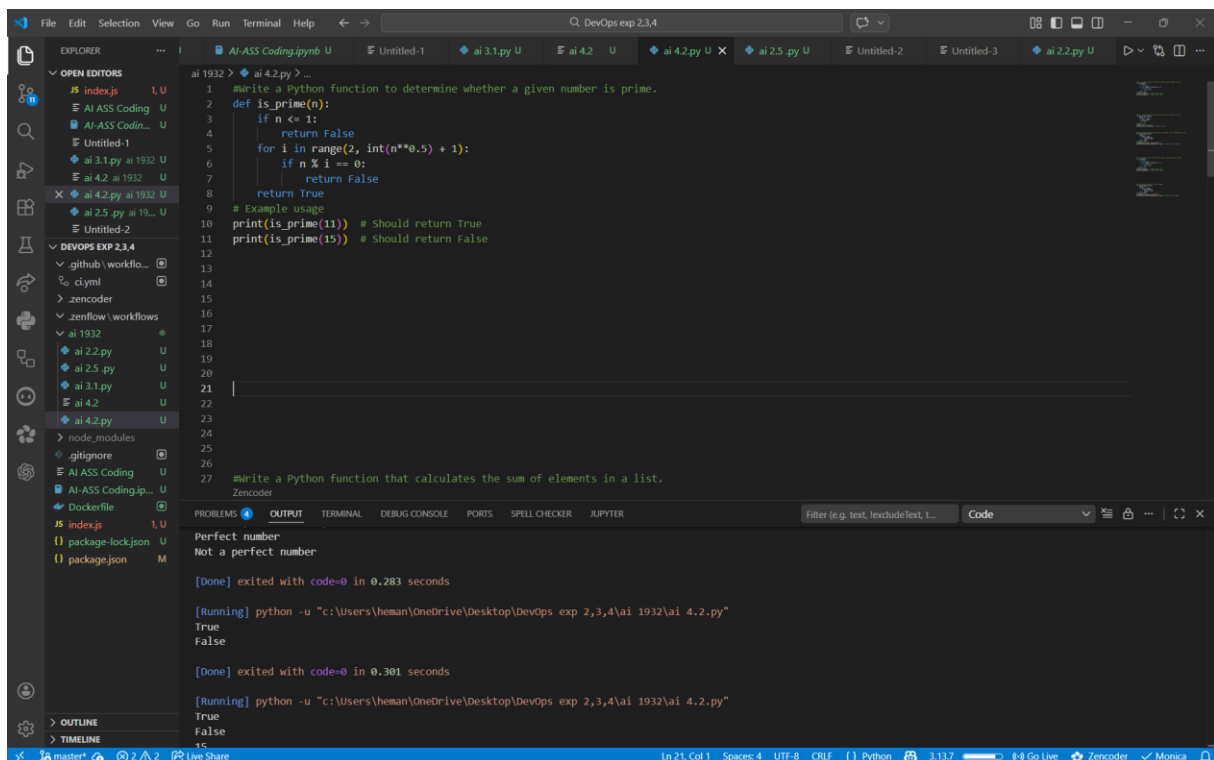
NAME-TEENESWARI

ROLLNO:2303A51932

TASK-1: ZERO-SHOT PROMPTING

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

CODE:



The screenshot shows a VS Code editor with a Python file named `ai 4.2.py`. The code defines a function `is_prime(n)` that checks if a number is prime. It includes example usage with `print(is_prime(11))` and `print(is_prime(15))`. The output pane shows the results: `Perfect number`, `Not a perfect number`, and the execution of the prime function returning `True` for 11 and `False` for 15.

```
1 #write a python function to determine whether a given number is prime.
2 def is_prime(n):
3     if n <= 1:
4         return False
5     for i in range(2, int(n**0.5) + 1):
6         if n % i == 0:
7             return False
8     return True
9
10 # Example usage
11 print(is_prime(11)) # Should return True
12 print(is_prime(15)) # Should return False
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27 #write a Python function that calculates the sum of elements in a list.
```

Perfect number
Not a perfect number
[Done] exited with code=0 in 0.283 seconds
[Running] python -u "c:\Users\heman\OneDrive\Desktop\DevOps exp 2,3,4\ai 1932\ai 4.2.py"
True
False
[Done] exited with code=0 in 0.301 seconds
[Running] python -u "c:\Users\heman\OneDrive\Desktop\DevOps exp 2,3,4\ai 1932\ai 4.2.py"
True
False

OBSERVATION:

- AI model understands the concept of a prime number without being given any examples or additional guidance -It applies correct mathematical reasoning purely from the instruction - The model generates syntactically correct and logically sound Python code

TASK-2

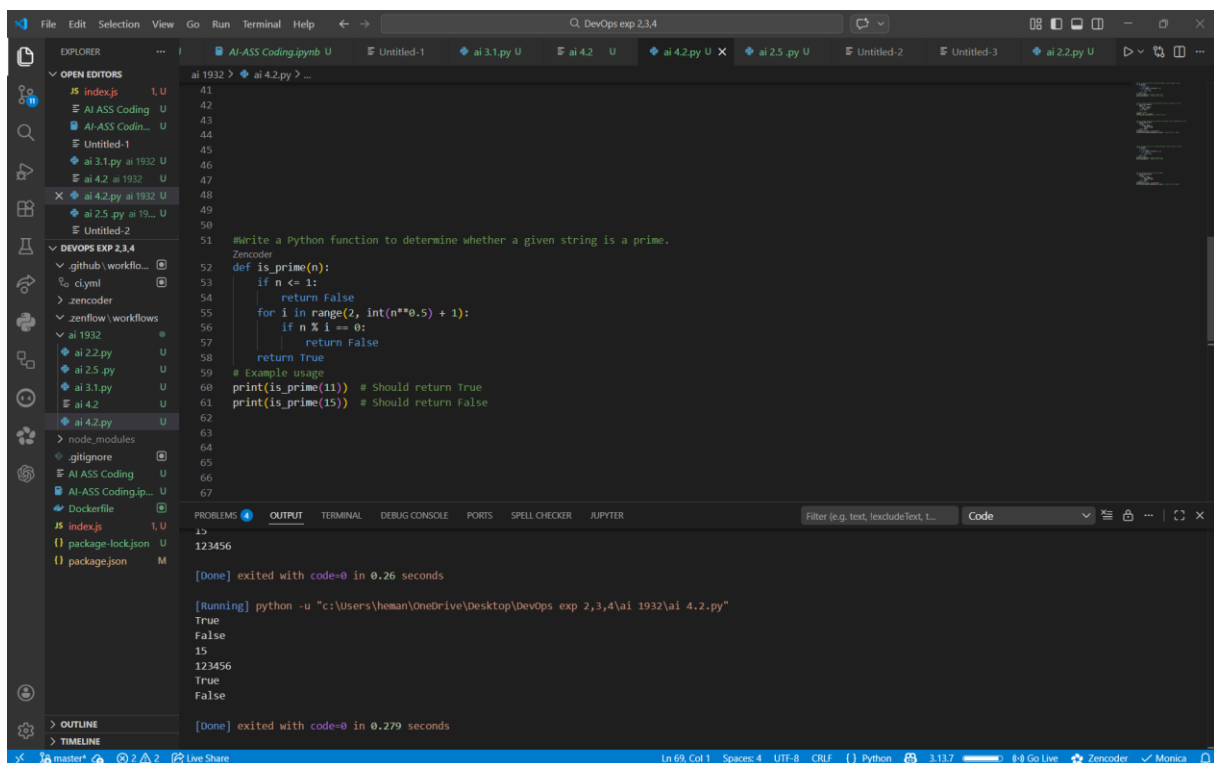
PROMPT: Write a Python function that calculates the sum of elements in a list.

Example:

Input: [1, 2, 3, 4]

Output: 10

CODE:



```
41
42
43
44
45
46
47
48
49
50
51 #Write a Python function to determine whether a given string is a prime.
52 def is_prime(n):
53     if n <= 1:
54         return False
55     for i in range(2, int(n**0.5) + 1):
56         if n % i == 0:
57             return False
58     return True
59 # Example usage
60 print(is_prime(11)) # Should return True
61 print(is_prime(15)) # Should return False
62
63
64
65
66
67
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE PORTS SPELL CHECKER JUPYTER

Filter (e.g. text, includeText, L... Code

[Done] exited with code=0 in 0.26 seconds

[Running] python -u "c:\Users\heman\OneDrive\Desktop\DevOps exp 2,3,4\ai 1932\ai 4.2.py"

True
False
15
123456
True
False

[Done] exited with code=0 in 0.279 seconds

OBSERVATION:

The single example guides the AI model to understand the expected input and output relationship The model correctly generalizes the pattern from the example to any list of numbers

TASK-3

PROMPT: Write a Python function that extracts only digits from an alphanumeric string.

Examples:

Input: "a1b2c3"

Output: "123"

Input: "x9y8z7"

Output: "987"

Input: "abc123def"

Output: "123" CODE:

```
25 numbers = [1, 2, 3, 4, 5]
26 print(sum_of_list(numbers)) # Should return 15
27
28
29
30
31 #Write a Python function that extracts only digits from an alphanumeric string.
32 def extract_digits(text):
33     result = ""
34     for char in text:
35         if char.isdigit():
36             result += char
37     return result
38 # Example usage
39 alphanumeric_string = "abc123def456"
40 print(extract_digits(alphanumeric_string)) # Should return '123456'
41
42
43
44
45
46
47
48
49
50
51 #Write a Python function to determine whether a given string is a prime.
```

```
15
[Done] exited with code=0 in 0.216 seconds

[Running] python -u "c:\Users\heman\OneDrive\Desktop\DevOps exp 2,3,4\ai 1932\ai 4.2.py"
True
False
15
123456

[Done] exited with code=0 in 0.26 seconds

[Running] python -u "c:\Users\heman\OneDrive\Desktop\DevOps exp 2,3,4\ai 1932\ai 4.2.py"
True
```

OBSERVATION:

- Multiple examples help the AI model clearly identify the pattern to be learned
- The model focuses only on digit characters and ignores alphabetic content
- The AI demonstrates improved confidence and reduced ambiguity compared to zero shot and one shot prompting

TASK-4

PROMPT: ZERO-SHOT: Write a Python function that counts the number of vowels in a string.

FEW-SHOT: Write a Python function that counts the number of vowels in a string.

Examples:

Input: "hello"

Output: 2

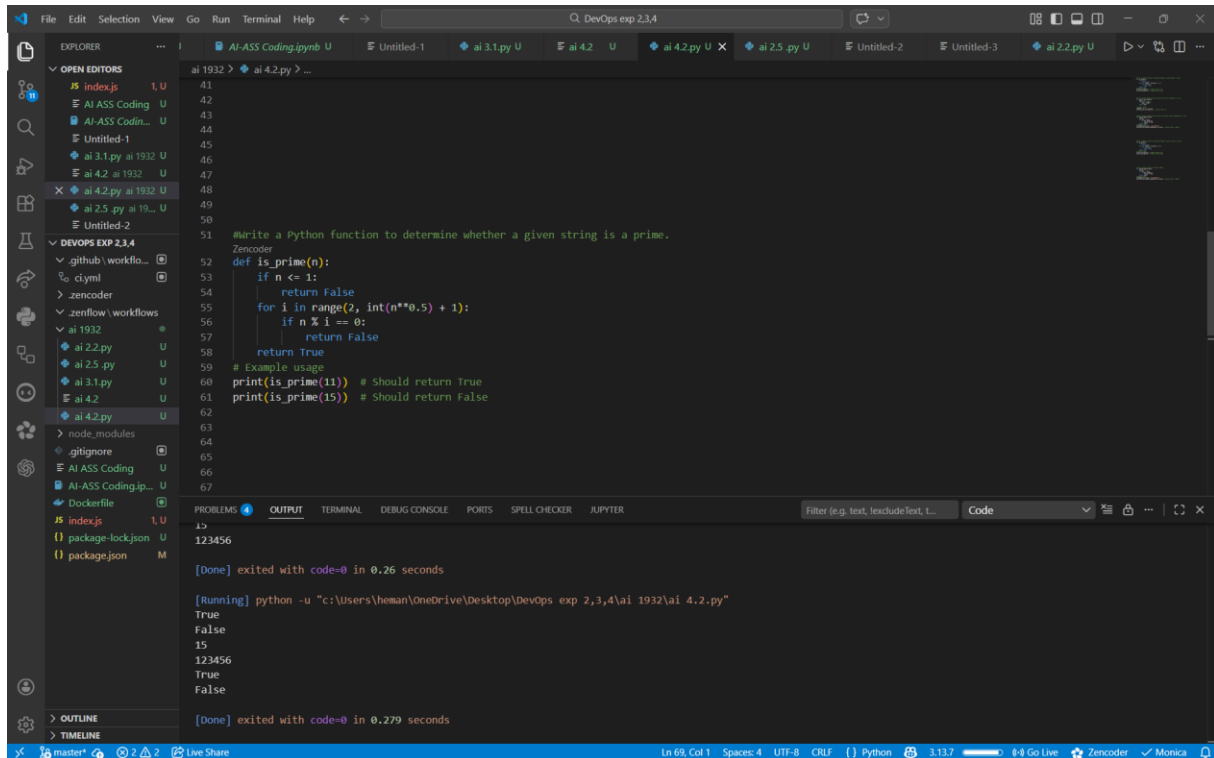
Input: "AEIOU"

Output: 5

Input: "chatgpt"

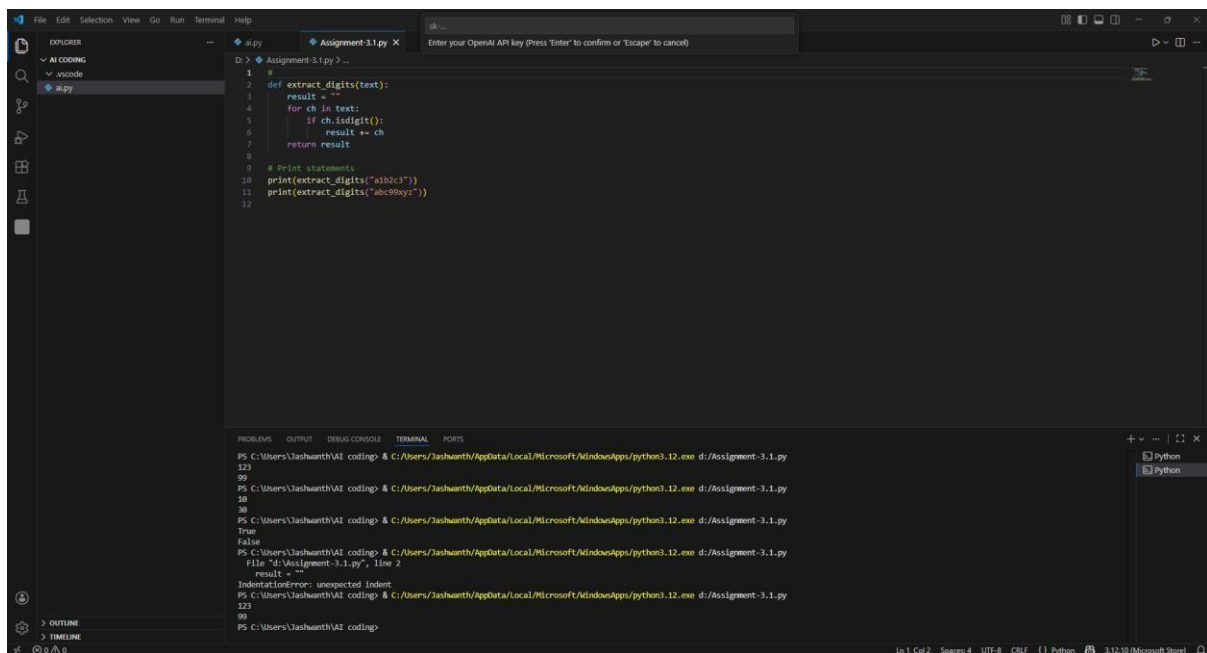
Output: 2 CODE:

ZERO-SHOT:



```
File Edit Selection View Go Run Terminal Help
Zencoder
ai 1932 > ai 4.2.py > ...
EXPLOSER
  OPEN EDITORS
    index.jp 1, U
    AI ASS Coding U
    AI ASS Codin... U
    Untitled-1
    ai 3.1.py ai 1932 U
    ai 4.2 ai 1932 U
    ai 4.2.py ai 1932 U
    ai 2.5.py ai 19... U
    Untitled-2
  DEVOPS EXP 2.3.4
    github\workflo...
    ciyaml
    zencoder
    zenflow\workflows
    ai 1932
    ai 2.2.py U
    ai 2.5.py U
    ai 3.1.py U
    ai 4.2 U
    ai 4.2.py U
    node_modules
    .gitignore
    AI ASS Coding U
    AI ASS Coding.jp... U
    Dockerfile
    index.jp 1, U
    package-lock.json U
    package.json M
  PROBLEMS
  OUTPUT
  TERMINAL
  DEBUG CONSOLE
  PORTS
  SPELL CHECKER
  JUPYTER
  Filter (e.g. text, text, text, L...)
  Code
  [Done] exited with code=0 in 0.26 seconds
[Running] python -u "c:\Users\heman\OneDrive\Desktop\DevOps exp 2,3,4\ai 1932\ai 4.2.py"
True
False
15
123456
True
False
[Done] exited with code=0 in 0.279 seconds
Ln 69, Col 1 Spaces: 4 UTF-8 CRLF Python 3.13.7 Go Live Zencoder Monica
```

FEW-SHOT:



```
File Edit Selection View Go Run Terminal Help
Assignment 3.1.py X
Enter your OpenAI API key (Press 'Enter' to confirm or 'Escape' to cancel)
1
2 def extract_digits(text):
3     result = ""
4     for ch in text:
5         if ch.isdigit():
6             result += ch
7     return result
8
9 # Print statements
10 print(extract_digits("a1b2c3"))
11 print(extract_digits("abc99xyz"))
12
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\jashwanth\AI coding> & C:\Users\jashwanth\AppData\Local\Microsoft\WindowsApps\python.12.exe d:/Assignment-3.1.py
123
99
PS C:\Users\jashwanth\AI coding> & C:\Users\jashwanth\AppData\Local\Microsoft\WindowsApps\python.12.exe d:/Assignment-3.1.py
True
False
PS C:\Users\jashwanth\AI coding> & C:\Users\jashwanth\AppData\Local\Microsoft\WindowsApps\python.12.exe d:/Assignment-3.1.py
File "d:\Assignment-3.1.py", line 2
    result = ""
    ^
IndentationError: unexpected indent
PS C:\Users\jashwanth\AI coding> & C:\Users\jashwanth\AppData\Local\Microsoft\WindowsApps\python.12.exe d:/Assignment-3.1.py
123
99
PS C:\Users\jashwanth\AI coding>
Ln 1, Col 2 Spaces: 4 UTF-8 CRLF Python 3.12.10 (Microsoft Store)
```

```
def count_vowels(text):  
    vowels = "aeiouAEIOU"  
    count = 0  
    for ch in text:  
        if ch in vowels:  
            count += 1  
    return count
```

OBSERVATION:

FEW-SHOT OBSERVATION

The provided examples clearly define what characters should be counted as vowels. The model confidently includes both uppercase and lowercase vowels due to examples. ZERO SHOT:

zero shot prompting the AI guesses the intent based on general knowledge which may vary for ambiguous tasks

TASK-5

PROMPT:

Write a Python function that determines the minimum of three numbers without using the built-in min() function.

Examples:

Input: 3, 7, 5

Output: 3

Input: 10, 2, 8

Output: 2

Input: 4, 4, 9

Output: 4 CODE:

The screenshot shows a Visual Studio Code editor window. The main editor displays a Python file named `Assignment 3.1.py` with the following code:

```
D:\> Assignment 3.1.py -
1 def find_min(a, b, c):
2     if a <= b and a <= c:
3         return a
4     elif b <= a and b <= c:
5         return b
6     else:
7         return c
8
9 # Print statements
10 print(find_min(3, 7, 5))
11 print(find_min(10, 2, 8))
12 print(find_min(4, 4, 9))
13
```

The bottom panel shows the `TERMINAL` output, which contains the following text:

```
PS C:\Users\Jashwanth\AI coding> & C:\Users\Jashwanth\AppData\Local\Microsoft\WindowsApps\python3.12.exe d:/Assignment-3.1.py
File "d:/Assignment-3.1.py", line 2
    result =
IndentationError: unexpected indent
PS C:\Users\Jashwanth\AI coding> & C:\Users\Jashwanth\AppData\Local\Microsoft\WindowsApps\python3.12.exe d:/Assignment-3.1.py
30
PS C:\Users\Jashwanth\AI coding> & C:\Users\Jashwanth\AppData\Local\Microsoft\WindowsApps\python3.12.exe d:/Assignment-3.1.py
True
False
PS C:\Users\Jashwanth\AI coding> & C:\Users\Jashwanth\AppData\Local\Microsoft\WindowsApps\python3.12.exe d:/Assignment-3.1.py
True
False
PS C:\Users\Jashwanth\AI coding> & C:\Users\Jashwanth\AppData\Local\Microsoft\WindowsApps\python3.12.exe d:/Assignment-3.1.py
3
2
4
PS C:\Users\Jashwanth\AI coding>
```

OBSERVATION:

The examples clearly establish the comparison pattern needed to identify the smallest value. The AI model infers the requirement to handle equality cases correctly. Conditional logic is generated without relying on built-in functions.