

ASSIGNMENT-4.2

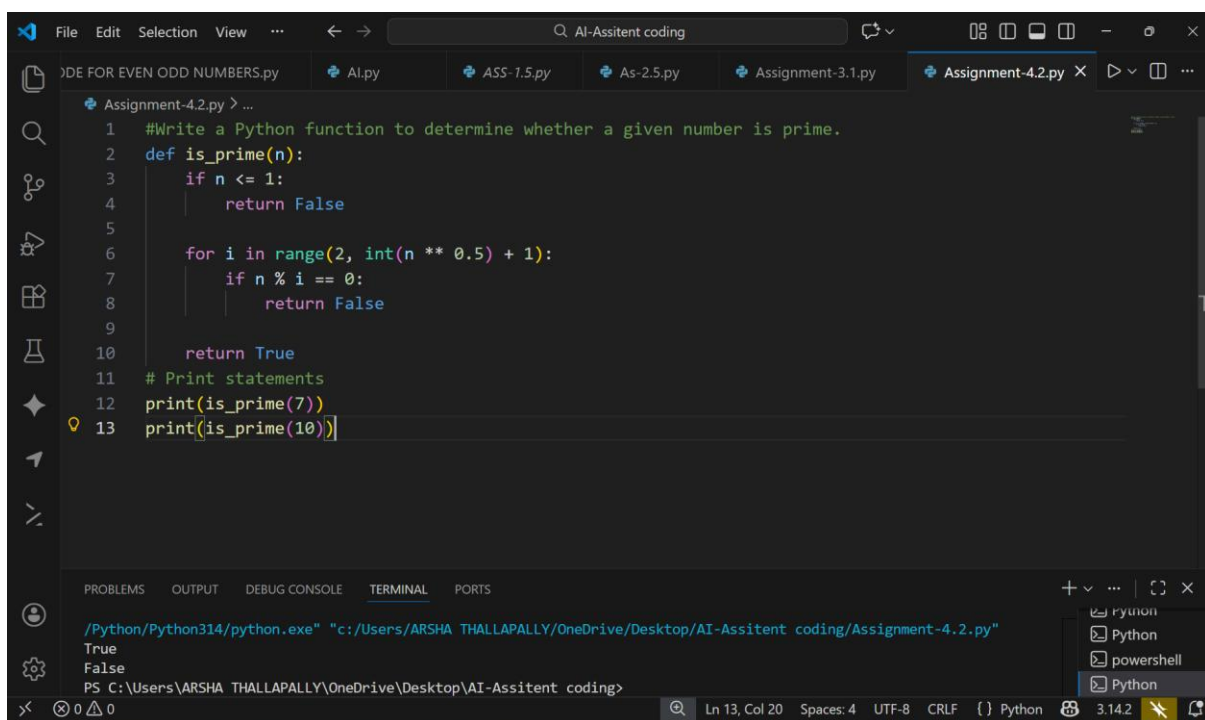
NAME-ARSHA VARDHINI

ROLL_NO:2303A51600

TASK-1: ZERO-SHOT PROMPTING

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

CODE:



```
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
6     for i in range(2, int(n ** 0.5) + 1):
7         if n % i == 0:
8             return False
9
10    return True
11
12 # Print statements
13 print(is_prime(7))
14 print(is_prime(10))
```

Terminal Output:

```
/Python/Python314/python.exe "c:/Users/ARSHA THALLAPALLY/OneDrive/Desktop/AI-Assitent coding/Assignment-4.2.py"
True
False
PS C:\Users\ARSHA THALLAPALLY\OneDrive\Desktop\AI-Assitent coding>
```

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

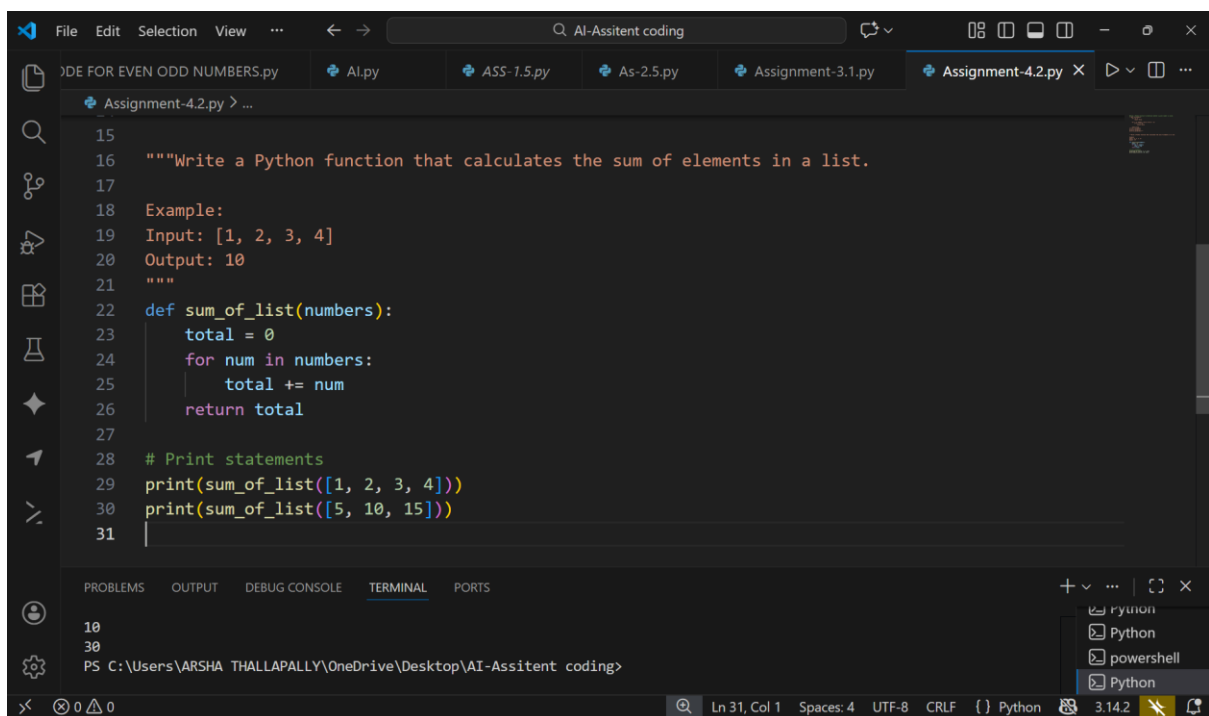
PROMPT: Write a Python function that calculates the sum of elements in a list.

Example:

Input: [1, 2, 3, 4]

Output: 10

CODE:

A screenshot of the Visual Studio Code (VS Code) editor interface. The main window displays a Python file named 'Assignment-4.2.py'. The code defines a function 'sum_of_list' that takes a list of numbers and returns their sum. The function is tested with two examples: [1, 2, 3, 4] and [5, 10, 15]. The output of the first test is 10, and the output of the second test is 30. The terminal at the bottom shows the execution of the code, displaying the outputs 10 and 30. The status bar at the bottom indicates the current position is Line 31, Column 1, with 4 spaces, using UTF-8 encoding and CRLF line endings, in a Python file.

```
15
16 """Write a Python function that calculates the sum of elements in a list.
17
18 Example:
19 Input: [1, 2, 3, 4]
20 Output: 10
21 """
22 def sum_of_list(numbers):
23     total = 0
24     for num in numbers:
25         total += num
26     return total
27
28 # Print statements
29 print(sum_of_list([1, 2, 3, 4]))
30 print(sum_of_list([5, 10, 15]))
31
```

10
30
PS C:\Users\ARSHA THALLAPALLY\OneDrive\Desktop\AI-Assitent coding>

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"

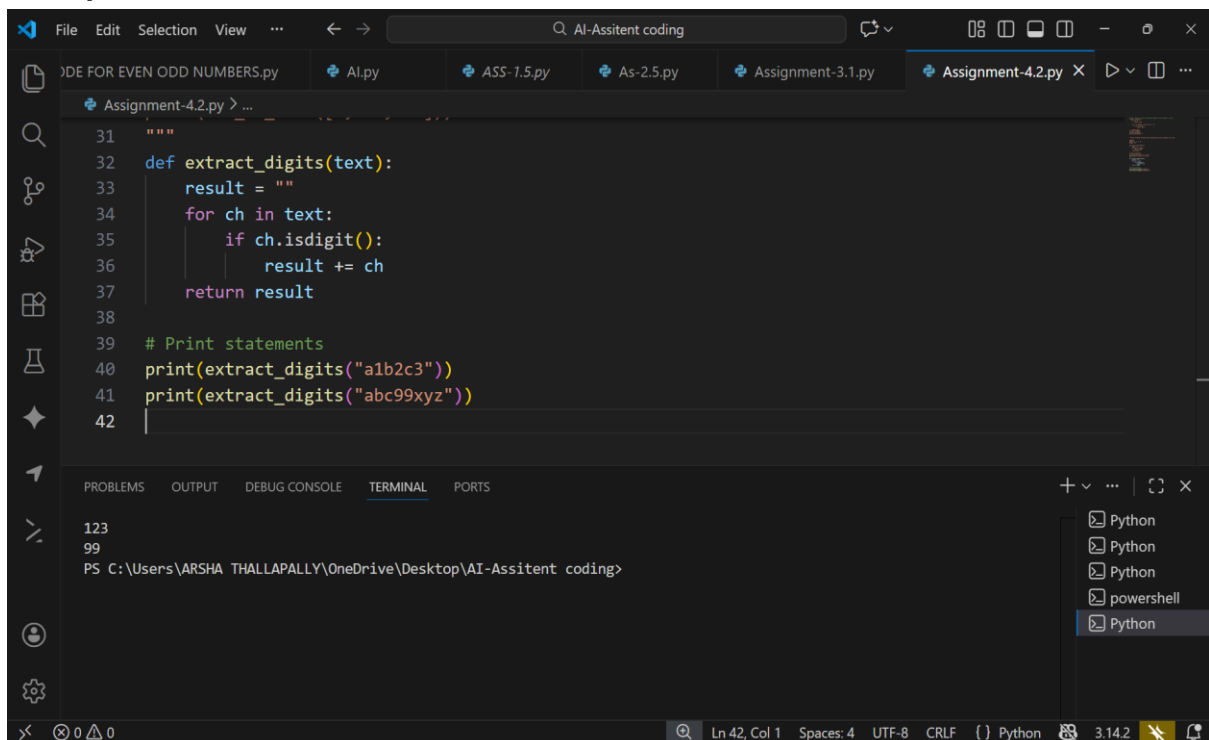
Output: "123"

Input: "x9y8z7"

Output: "987"

Input: "abc123def"

Output: "123"



The screenshot shows a Visual Studio Code editor window with a Python file named 'Assignment-4.2.py'. The code defines a function 'extract_digits(text)' that iterates through each character in the string and appends it to a 'result' string if it is a digit. Below the function, there are two print statements: 'print(extract_digits("a1b2c3"))' and 'print(extract_digits("abc99xyz"))'. The terminal at the bottom shows the output of these print statements: '123' and '99'. The terminal prompt is 'PS C:\Users\ARSHA THALLAPALLY\OneDrive\Desktop\AI-Assitent coding>'. The status bar at the bottom indicates the current line and column is 'Ln 42, Col 1', the file is encoded in 'UTF-8', and the Python version is '3.14.2'.

```
31 """
32 def extract_digits(text):
33     result = ""
34     for ch in text:
35         if ch.isdigit():
36             result += ch
37     return result
38
39 # Print statements
40 print(extract_digits("a1b2c3"))
41 print(extract_digits("abc99xyz"))
42
```

123
99
PS C:\Users\ARSHA THALLAPALLY\OneDrive\Desktop\AI-Assitent coding>

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"

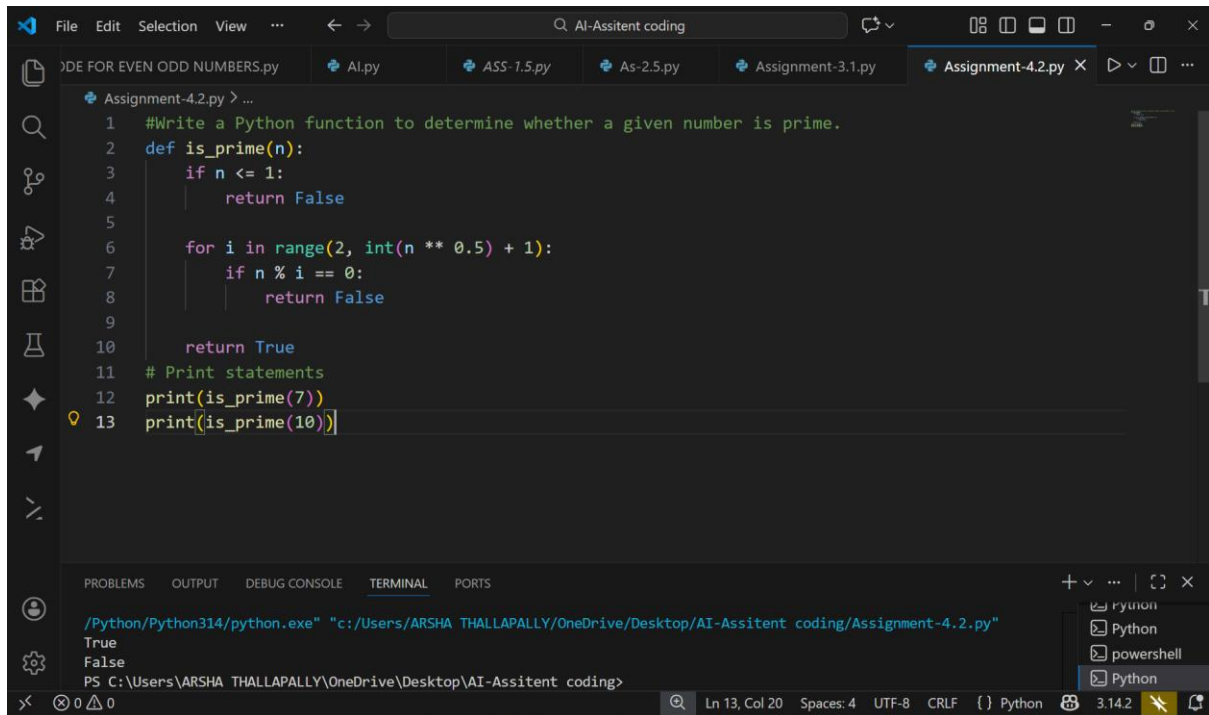
Output: 5

Input: "chatgpt"

Output: 2

CODE:

ZERO-SHOT:



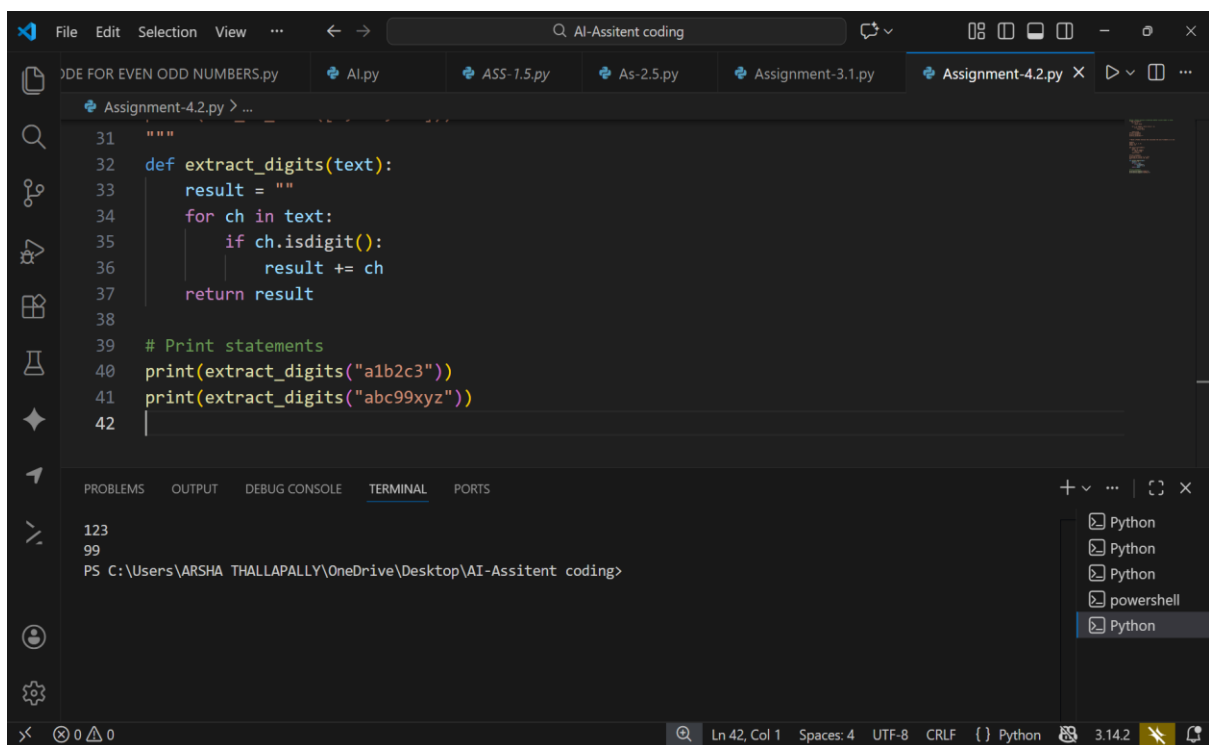
The screenshot shows a Visual Studio Code editor with a file named 'Assignment-4.2.py'. The code defines a function 'is_prime(n)' that checks if a number is prime. It returns False for n <= 1 and for numbers divisible by any integer from 2 to the square root of n. It then prints the results for 7 and 10. The terminal output shows 'True' for 7 and 'False' for 10.

```
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
6     for i in range(2, int(n ** 0.5) + 1):
7         if n % i == 0:
8             return False
9
10    return True
11 # Print statements
12 print(is_prime(7))
13 print(is_prime(10))
```

Terminal Output:

```
/Python/Python314/python.exe "c:/Users/ARSHA THALLAPALLY/OneDrive/Desktop/AI-Assitent coding/Assignment-4.2.py"
True
False
PS C:\Users\ARSHA THALLAPALLY\OneDrive\Desktop\AI-Assitent coding>
```

FEW-SHOT:



The screenshot shows a Visual Studio Code editor with a file named 'Assignment-4.2.py'. The code defines a function 'extract_digits(text)' that iterates through each character in the string and appends it to a result string if it is a digit. It then prints the results for 'a1b2c3' and 'abc99xyz'. The terminal output shows '123' and '99'.

```
31 """
32 def extract_digits(text):
33     result = ""
34     for ch in text:
35         if ch.isdigit():
36             result += ch
37     return result
38
39 # Print statements
40 print(extract_digits("a1b2c3"))
41 print(extract_digits("abc99xyz"))
42 |
```

Terminal Output:

```
123
99
PS C:\Users\ARSHA THALLAPALLY\OneDrive\Desktop\AI-Assitent coding>
```

```
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

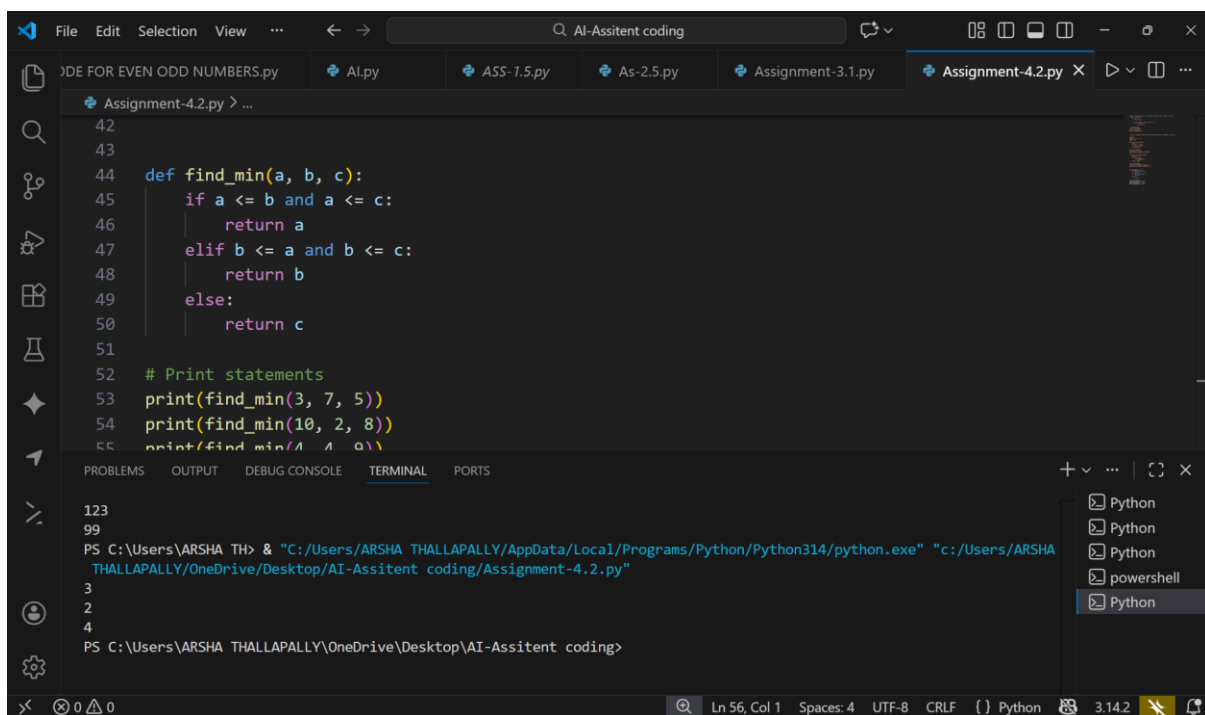
Input: 10, 2, 8

Output: 2

Input: 4, 4, 9

Output: 4

CODE:



The screenshot shows a Python IDE with a file named 'Assignment-4.2.py'. The code defines a function 'find_min(a, b, c)' that returns the minimum of three numbers using conditional logic. It then prints the results for three different input sets: (3, 7, 5), (10, 2, 8), and (4, 4, 9). The terminal output shows the function being called and the correct minimum values (3, 2, and 4) being printed. A context menu is open over the terminal, showing options for Python, powershell, and other shells.

```
42
43
44 def find_min(a, b, c):
45     if a <= b and a <= c:
46         return a
47     elif b <= a and b <= c:
48         return b
49     else:
50         return c
51
52 # Print statements
53 print(find_min(3, 7, 5))
54 print(find_min(10, 2, 8))
55 print(find_min(4, 4, 9))

123
99
PS C:\Users\ARSHA TH> & "C:/Users/ARSHA THALLAPALLY/AppData/Local/Programs/Python/Python314/python.exe" "c:/Users/ARSHA THALLAPALLY/OneDrive/Desktop/AI-Assitent coding/Assignment-4.2.py"
3
2
4
PS C:\Users\ARSHA THALLAPALLY\OneDrive\Desktop\AI-Assitent 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

