AI ASSISTED CODING

NAME: Abhilash

HALL_N0: 2303A51227

Question 1: Zero-Shot Prompting (Palindrome Number Program)

Write a zero-shot prompt (without providing any examples) to generate

a Python function that checks whether a given number is a palindrome.

Task:

- Record the AI-generated code.
- Test the code with multiple inputs.
- Identify any logical errors or missing edge-case handling.

CODE:

#write a program to check a number is palindrome

```python
#write a program to check a number is palindrome
num = int(input("Enter a number: "))
temp = num
reverse = 0
while num > 0:
    digit = num % 10
    reverse = reverse * 10 + digit
    num = num // 10
if temp == reverse:
    print(f"{temp} is a palindrome")
else:
    print(f"{temp} is not a palindrome")
```

OUTPUT:

```
Enter a number: 121
121 is a palindrome
PS C:\Users\nithi\Desktop\ai assisted coding>
```

Question 2: One-Shot Prompting (Factorial Calculation)

Write a one-shot prompt by providing one input-output example and

ask the AI to generate a Python function to compute the factorial of a

given number.

Example:

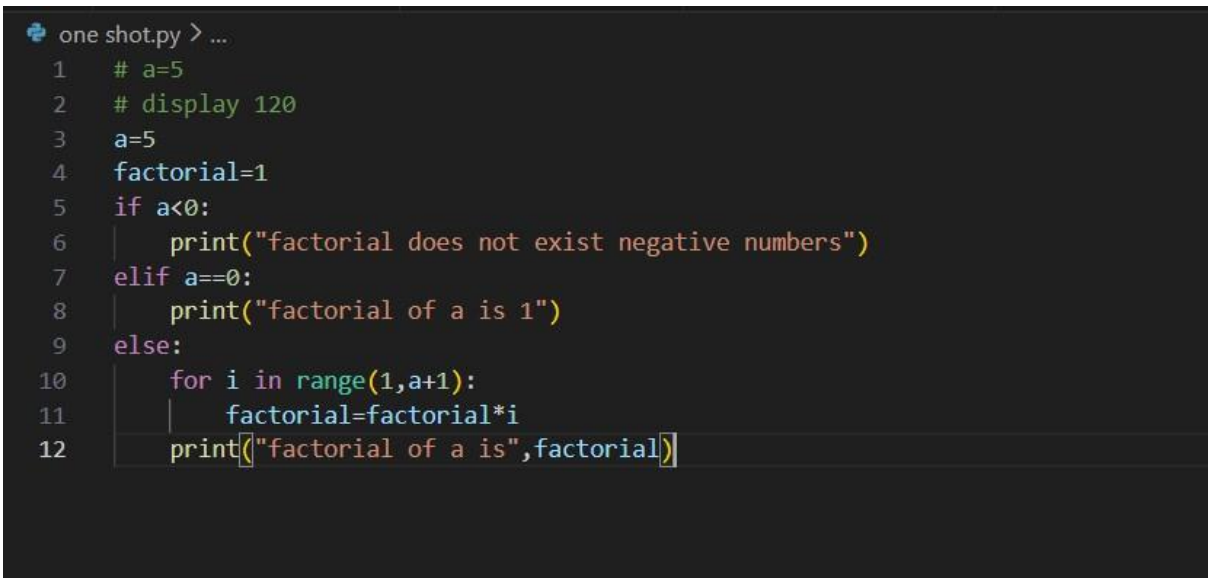
Input: 5 → Output: 120

Task:

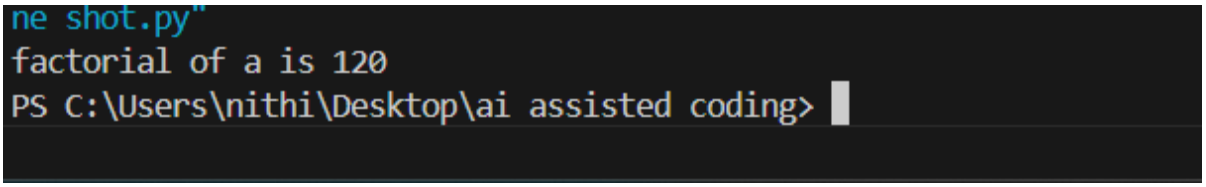
• Compare the generated code with a zero-shot solution.

• Examine improvements in clarity and correctness.

CODE:



Output:



Question 3: Few-Shot Prompting (Armstrong Number Check)

Write a few-shot prompt by providing multiple input-output examples

to guide the AI in generating a Python function to check whether a

given number is an Armstrong number.

Examples:

• Input: 153 → Output: Armstrong Number

• Input: 370 → Output: Armstrong Number

• Input: 123 → Output: Not an Armstrong Number

Task:

• Analyze how multiple examples influence code structure and

accuracy.

• Test the function with boundary values and invalid inputs.

Code:

```
#a=153
#display amstrong nuber
#a=370
#display amstrong nuber
#a=123
#display not an amstrong nuber
#write a python program to check whether a number is amstrong or not
a=int(input("enter a number:"))
sum=0
temp=a
while temp>0:
    digit=temp%10
    sum+=digit**3
    temp//=10
if a==sum:
    print(f"{a} is an amstrong number")
```

Input/outputs:

```
enter a number:153
153 is an amstrong number
PS C:\Users\nithi\Desktop\ai assisted coding>
```

Question 4: Context-Managed Prompting (Optimized Number

Classification)

Design a context-managed prompt with clear instructions and

constraints to generate an optimized Python program that classifies a

number as prime, composite, or neither.

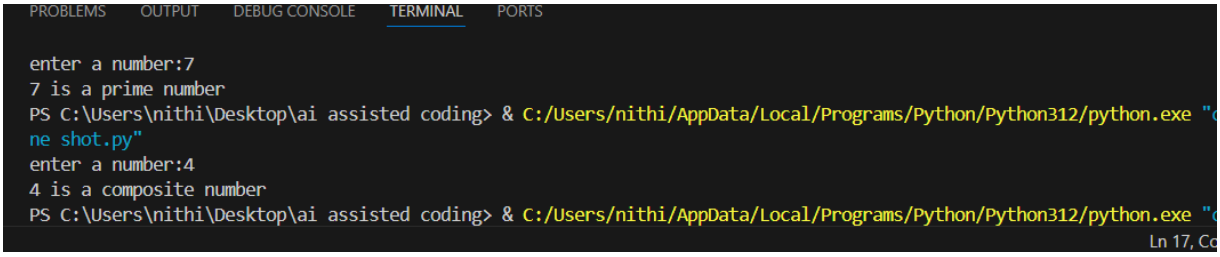
Task:

- Ensure proper input validation.
- Optimize the logic for efficiency.
- Compare the output with earlier prompting strategies.

Code:

```python
# n=7
# display is an prime
# n=4
# display is an composite
# n=1
# display neither prime nor composite
# write a python program to check whether a number is prime or composite
n=int(input("enter a number:"))
if n==1:
    print(f"{n} is neither prime nor composite")
elif n>1:
    for i in range(2,n):
        if n%i==0:
            print(f"{n} is a composite number")
            break
    else:
        print(f"{n} is a prime number")
```

Inputs/output:



Question 5: Zero-Shot Prompting (Perfect Number Check)

Write a zero-shot prompt (without providing any examples) to

generate a Python function that checks whether a given number is a

perfect number.

Task:

- Record the AI-generated code.
- Test the program with multiple inputs.
- Identify any missing conditions or inefficiencies in the logic.

Code:

```python
#write a python program to check whether a number is perfect number or not
n=int(input("enter a number:"))
sum=0
for i in range(1,n):
    if n%i==0:
        sum+=i
if sum==n:
    print(f"{n} is a perfect number")
else:
    print(f"{n} is not a perfect number")
```

Inputs/outputs:

```
enter a number:6
6 is a perfect number
PS C:\Users\nithi\Desktop\ai assisted coding> & C:/Users/nithi/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/nithi/Desktop/ai a
ne shot.py"
enter a number:18
18 is not a perfect number
PS C:\Users\nithi\Desktop\ai assisted coding>
```

Question 6: Few-Shot Prompting (Even or Odd Classification with

Validation)

Write a few-shot prompt by providing multiple input-output

examples to guide the AI in generating a Python program that

determines whether a given number is even or odd, including proper

input validation.

Examples:

• Input: 8 → Output: Even

• Input: 15 → Output: Odd

• Input: 0 → Output: Even

Task:

• Analyze how examples improve input handling and output

clarity.

• Test the program with negative numbers and non-integer inputs.

Code:

few shot.py > ...

```python
# n=8
# display is an even
# n=15
# display is an odd
# n=0
# display is an even
# write a python program to check whether a number is even or odd
n=int(input("enter a number:"))
if n%2==0:
    print(f"{n} is an even number")
else:
    print(f"{n} is an odd number")
```

Inputs/outputs:

```
enter a number:8
8 is an even number
PS C:\Users\nithi\Desktop\ai assisted coding> & C:/Users/nithi/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/n
ew shot.py"
enter a number:15
15 is an odd number
PS C:\Users\nithi\Desktop\ai assisted coding>
```