

AI ASSISTED CODING

NAME: NITHIN

HALL_N0: 2303A51096

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
```

```
1 #write a program to check a number is palindrome
2 num = int(input("Enter a number: "))
3 temp = num
4 reverse = 0
5 while num > 0:
6     digit = num % 10
7     reverse = reverse * 10 + digit
8     num = num // 10
9 if temp == reverse:
10    print(f"{temp} is a palindrome")
11 else:
12    print(f"{temp} is not a palindrome")
```

OUTPUT:

```
!nc -l -p 5000
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:

```
one shot.py > ...
1 # a=5
2 # display 120
3 a=5
4 factorial=1
5 if a<0:
6     print("factorial does not exist negative numbers")
7 elif a==0:
8     print("factorial of a is 1")
9 else:
10    for i in range(1,a+1):
11        factorial=factorial*i
12    print("factorial of a is",factorial)
```

Output:

```
ne shot.py"
factorial of a is 120
PS C:\Users\nithi\Desktop\ai assisted coding>
```

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:

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

Input/outputs:

```
nc_snoe.py
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:

```

# 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:

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

enter a number:7
7 is a prime number
PS C:\Users\nithi\Desktop\ai assisted coding> & C:/Users/nithi/AppData/Local/Programs/Python/Python312/python.exe "c:\Users\nithi\Desktop\ai assisted coding\ne_shot.py"
enter a number:4
4 is a composite number
PS C:\Users\nithi\Desktop\ai assisted coding> & C:/Users/nithi/AppData/Local/Programs/Python/Python312/python.exe "c:\Users\nithi\Desktop\ai assisted coding\ne_shot.py"
Ln 17, Col 1

```

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:

```
1  #write a python program to check whether a number is perfect number or not
2  n=int(input("enter a number:"))
3  sum=0
4  for i in range(1,n):
5      if n%i==0:
6          sum+=i
7  if sum==n:
8      print(f"{n} is a perfect number")
9  else:
10     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 /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 > ...
1  # n=8
2  # display is an even
3  # n=15
4  # display is an odd
5  # n=0
6  # display is an even
7  # write a python program to check whether a number is even or odd
8 n=int(input("enter a number:"))
9 if n%2==0:
10    print(f"{n} is an even number")
11 else:
12    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/nithi/Desktop/ai assisted coding> \shot.py"
enter a number:15
15 is an odd number
PS C:\Users\nithi\Desktop\ai assisted coding> []
```