

ASSIGNMENT-3.1

Name: K. Nithish Raj B-14

Task1:Zero-ShotPrompting(PalindromeNumberProgram)

Prompt:write a python function to check given number is palindrome or not return output in True or False. Provide code in user input.

Codewith output:

```
zeroshot(palindrome).py > ...
1  #write a python function to check given number is palindrome or not return output in True or False.provide code in user input
2  def is_palindrome(number):
3      # Convert the number to string
4      str_num = str(number)
5      # Check if the string is equal to its reverse
6      return str_num == str_num[::-1]
7  # Get user input
8  user_input = int(input("Enter a number: "))
9  # Check if the number is palindrome and print the result
10 result = is_palindrome(user_input)
11 print(result)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\akhil\OneDrive\Documents\AI(vs)> & C:/Users/akhil/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/akhil/OneDrive/Docum
ents/AI(vs)/oneshot(factorial).py"
Enter a number: 5
120
PS C:\Users\akhil\OneDrive\Documents\AI(vs)> & C:/Users/akhil/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/akhil/OneDrive/Docum
ents/AI(vs)/oneshot(factorial).py"
Enter a number: 9
362880
PS C:\Users\akhil\OneDrive\Documents\AI(vs)> & C:/Users/akhil/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/akhil/OneDrive/Docum
ents/AI(vs)/oneshot(factorial).py"
Enter a number: 4
24
```

Justification:

- Correct for positive numbers
- Negative numbers not handled clearly
- No input validation for non-integers

Task2:One-ShotPrompting(FactorialCalculation)

Prompt:write a python function to calculate factorial of given number. Example: input=5; output=120

Codewith output:

```
oneshot(factorial).py > ...
1  #write a python function to calculate factorial of given number. Example: input=5 ; output=120
2  def factorial(n):
3      # Initialize result
4      result = 1
5      # Calculate factorial
6      for i in range(1, n + 1):
7          result *= i
8      return result
9  # Get user input
10 user_input = int(input("Enter a number: "))
11 # Calculate factorial and print the result
12 result = factorial(user_input)
13 print(result)]

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS C:\Users\akhil\OneDrive\Documents\AI(vs)> & C:/Users/akhil/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/akhil/OneDrive/Docum
ents/AI(vs)/oneshot(factorial).py"
Enter a number: 5
120
PS C:\Users\akhil\OneDrive\Documents\AI(vs)> & C:/Users/akhil/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/akhil/OneDrive/Docum
ents/AI(vs)/oneshot(factorial).py"
Enter a number: 9
362880
PS C:\Users\akhil\OneDrive\Documents\AI(vs)> & C:/Users/akhil/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/akhil/OneDrive/Docum
ents/AI(vs)/oneshot(factorial).py"
Enter a number: 4
24
PS C:\Users\akhil\OneDrive\Documents\AI(vs)> ]
```

Justification:

- One-shot gives clearer logic
- More readable and reliable

Task3: Few-Shot Prompting (Armstrong Number Check) Code

with output:

```
fewshot(armstrong).py > ...
1 #Write a python function to check given number is an armstrong number.
2 # Example: input=153 ; output=Armstrong Number , input=123 ; output=Not an Armstrong Number
3
4 def is_armstrong_number(number):
5     # Convert the number to string
6     str_num = str(number)
7     # Calculate the number of digits
8     num_digits = len(str_num)
9     # Calculate the sum of each digit raised to the power of the number of digits
10    sum_of_powers = sum(int(digit) ** num_digits for digit in str_num)
11    # Check if the sum equals the original number
12    return sum_of_powers == number
13
14 # Get user input
15 user_input = int(input("Enter a number: "))
16
17 # Check if the number is an Armstrong number and print the result
18 result = is_armstrong_number(user_input)
19 if result:
20     print("Armstrong Number")
21 else:
22     print("Not an Armstrong Number")

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\akhil\OneDrive\Documents\AI(vs)> & C:/Users/akhil/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/akhil/OneDrive/Documents/AI(vs)/fewshot(armstrong).py"
Enter a number: 123
Not an Armstrong Number
PS C:\Users\akhil\OneDrive\Documents\AI(vs)> & C:/Users/akhil/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/akhil/OneDrive/Documents/AI(vs)/fewshot(armstrong).py"
Enter a number: 153
Armstrong Number
PS C:\Users\akhil\OneDrive\Documents\AI(vs)> & C:/Users/akhil/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/akhil/OneDrive/Documents/AI(vs)/fewshot(armstrong).py"
Enter a number: 370
Armstrong Number
```

Justification:

- Few-shot improves accuracy
- Code structure matches examples
- Still lacks input validation for negatives

Task4: Context-Managed Prompting (Optimized Number

Classification)

Prompt: generate an optimized python program that classifies a number as prime, composite or neither

Codewith output:

