AIAC LAB

Lab Assignment- 3.1

Experiment – Prompt Engineering Techniques

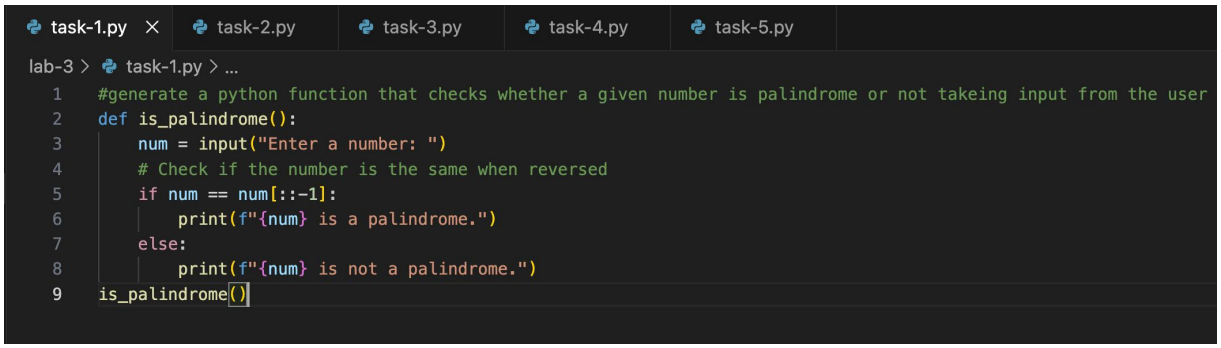
2303A51721

Batch-11

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

Prompts: Generate a Python function that checks whether a given number is a palindrome.

Code:



Output:

Explaination: This code determines whether a number entered by the user is a palindrome, which means it reads the same forwards and backwards. When the user inputs a number, the code first converts it to a string so that it can easily compare the sequence of digits. It then checks if this string is identical to its reverse. If they match, the function returns True, indicating the number is a palindrome; otherwise, it returns False. The result is printed out, letting the user know whether their number is a palindrome or not.

Question 2: One-Shot Prompting (Factorial Calculation)

Prompts: Generate a Python function to compute the factorial of a given number.

Code:

task-1.py | task-2.py | task-3.py | task-4.py | task-5.py

lab-3 > task-2.py > ...

```python
#generate a python function to compute the factorial of a given number
def factorial():
    num = int(input("Enter a number: "))
    result = 1
    for i in range(1, num + 1):
        result *= i
    print(f"The factorial of {num} is {result}.")
factorial()
```

Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS  Python

/usr/local/bin/python3 "/Users/gandirakesh/Desktop/aiac lab assignments/lab-3/task-2.py"
(base) gandirakesh@Gandis-MacBook-Air-2 aiac lab assignments % /usr/local/bin/python3 "/Users/gandirakesh/Desktop/aiac lab assignments/lab-3/task-2.py"
Enter a number: 6
The factorial of 6 is 720.
(base) gandirakesh@Gandis-MacBook-Air-2 aiac lab assignments %
```

Explaination: This code defines a function to calculate the factorial of a given number using recursion. The factorial of a number (written as n!) is the product of all positive integers up to that number. For example, 5! = 5 × 4 × 3 × 2 × 1 = 120. The function checks if the input is 0 or 1, in which case it returns 1, since the factorial of both 0 and 1 is 1. The example usage shows how the function works for different inputs, printing the results for 5, 0, and 3.

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

Prompts: Generating a Python function to check whether a given number is an Armstrong number.

Code:

lab-3 > task-3.py > ...

```python
#generate a python function to checl whether the given number is armstrong number or not
def is_armstrong():
    num = int(input("Enter a number: "))
    # Calculate the number of digits
    num_digits = len(str(num))
    sum_of_powers = 0
    temp = num
    while temp > 0:
        digit = temp % 10
        sum_of_powers += digit ** num_digits
        temp //= 10
    if sum_of_powers == num:
        print(f"{num} is an Armstrong number.")
    else:
        print(f"{num} is not an Armstrong number.")
is_armstrong()
```

Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS                Python
/usr/local/bin/python3 "/Users/gandirakesh/Desktop/aiac lab assignments/lab-3/task-3.py"
(base) gandirakesh@Gandis-MacBook-Air-2 aiac lab assignments % /usr/local/bin/python3 "/Users/gandirakesh/Desktop/aiac
 lab assignments/lab-3/task-3.py"
Enter a number: 65
65 is not an Armstrong number.
(base) gandirakesh@Gandis-MacBook-Air-2 aiac lab assignments %
```

Explaination: This code defines a function to check if a number is an Armstrong number. An Armstrong number is a number that is equal to the sum of its own digits each raised to the power of the number of digits. The function works by first converting the number to a string so it can easily access each digit. It then calculates the number of digits and computes the sum of each digit raised to that power. If this sum matches the original number, the function returns True; otherwise, it returns False

Question 4: Context-Managed Prompting (Optimized Number Classification)

Prompts: Generate an optimized Python program that classifies a number as prime, composite, or neither.

Code:

task-1.py | task-2.py | task-3.py | task-4.py | task-5.py

lab-3 > task-4.py > ...

```python
#Generate an optimized Python program that classifies a number as prime, composite, or neither
def classify_number():
    num = int(input("Enter a number: "))
    if num <= 1:
        print(f"{num} is neither prime nor composite.")
        return
    for i in range(2, int(num**0.5) + 1):
        if num % i == 0:
            print(f"{num} is a composite number.")
            return
    print(f"{num} is a prime number.")
classify_number()
```

Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS                Python
/usr/local/bin/python3 "/Users/gandirakesh/Desktop/aiac lab assignments/lab-3/task-4.py"
(base) gandirakesh@Gandis-MacBook-Air-2 aiac lab assignments % /usr/local/bin/python3 "/Users/gandirakesh/Desktop/aiac
 lab assignments/lab-3/task-4.py"
Enter a number: 23
23 is a prime number.
(base) gandirakesh@Gandis-MacBook-Air-2 aiac lab assignments %
```

Explaination: This code defines a function that classifies a given number as "prime," "composite," or "neither." If the input number is less than or equal to 1, it returns "neither" because 0 and 1 are not considered prime or composite. For numbers greater than 1, it checks if the number is divisible by any integer from 2 up to the square root of the number. If it finds a divisor, the number is "composite".

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

Prompts: Generate a Python function that checks whether a given number is a perfect number.

Code:

task-1.py | task-2.py | task-3.py | task-4.py | task-5.py

lab-3 > task-5.py > ...

```python
#Generate a Python function that checks whether a given number is a perfect number
def is_perfect_number():
    num = int(input("Enter a number: "))
    divisors_sum = 0
    for i in range(1, num):
        if num % i == 0:
            divisors_sum += i
    if divisors_sum == num:
        print(f"{num} is a perfect number.")
    else:
        print(f"{num} is not a perfect number.")
is_perfect_number()
```

Output:

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS Python

```
/usr/local/bin/python3 "/Users/gandirakesh/Desktop/aiac lab assignments/lab-3/task-5.py"
(base) gandirakesh@Gandis-MacBook-Air-2 aiac lab assignments % /usr/local/bin/python3 "/Users/gandirakesh/Desktop/aiac lab assignments/lab-3/task-5.py"
Enter a number: 23
23 is not a perfect number.
(base) gandirakesh@Gandis-MacBook-Air-2 aiac lab assignments %
```

Explaination: This code defines a function to check if a number is a perfect number. A perfect number is a positive integer that is equal to the sum of all its proper divisors (excluding itself). The function first checks if the number is less than 1, returning False since perfect numbers must be positive. It then calculates the sum of all divisors of the number from 1 up to (but not including) the number itself. If this sum equals the original number, the function returns True.

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

Prompts: Determines whether a given number is even or odd, including proper input validation.

Code:

task-1.py | task-2.py | task-3.py | task-4.py | task-5.py | task-6.py

lab-3 > task-6.py > ...

```python
#Determines whether a given number is even or odd, including proper input validation
def is_even_or_odd():
    while True:
        try:
            num = int(input("Enter a number: "))
            break
        except ValueError:
            print("Invalid input. Please enter an integer.")
    if num % 2 == 0:
        print(f"{num} is even.")
    else:
        print(f"{num} is odd.")
is_even_or_odd()
```

Output:

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS Python

```
/usr/local/bin/python3 "/Users/gandirakesh/Desktop/aiac lab assignments/lab-3/task-6.py"
(base) gandirakesh@Gandis-MacBook-Air-2 aiac lab assignments % /usr/local/bin/python3 "/Users/gandirakesh/Desktop/aiac lab assignments/lab-3/task-6.py"
Enter a number: 765
765 is odd.
(base) gandirakesh@Gandis-MacBook-Air-2 aiac lab assignments %
```

Explaination: This code defines a function that determines whether a given number is even or odd, with input validation. The function first checks if the input is an integer; if not, it returns a message asking for a valid integer. If the input is an integer, it checks if the number is divisible by 2. If it is, the function returns "even"; otherwise, it returns "odd."