AI ASSITANT CODING LAB-3.2

- Name:**K.Sathvik**
- H.T NO:**2303A51202**
- Batch.NO:**04**

Task 1-Progressive Calculator

Progressive Prompting for Calculator Design: Ask the AI to design a simple calculator program by initially providing only the function name. Gradually enhance the prompt by adding comments and usage examples.

```python
#write a program to design a simple calculator
def add(x, y):
    return x + y

def subtract(x, y):
    return x - y

def multiply(x, y):
    return x * y

def divide(x, y):
    if y == 0:
        return "Error! Division by zero."
    return x / y

def power(x, y):
    return x ** y

print("Select operation:")
print("1. Addition")
print("2. Subtraction")
print("3. Multiplication")
print("4. Division")
print("5. Power")

choice = input("Enter choice(1/2/3/4/5): ")
num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))

if choice == '1':
    print(f"{num1} + {num2} = {add(num1, num2)}")
elif choice == '2':
    print(f"{num1} - {num2} = {subtract(num1, num2)}")
elif choice == '3':
    print(f"{num1} * {num2} = {multiply(num1, num2)}")
elif choice == '4':
    print(f"{num1} / {num2} = {divide(num1, num2)}")
elif choice == '5':
    print(f"{num1} ^ {num2} = {power(num1, num2)}")
else:
    print("Invalid input")
```

```
Select operation:
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Power
Enter choice(1/2/3/4/5): 2
Enter first number: 6
Enter second number: 4
6.0 - 4.0 = 2.0
```

Task-2:Sorting Student Marks

Refining Prompts for Sorting Logic: Start with a vague prompt for sorting student marks, then refine it to clearly specify sorting order and constraints.

```python
def sort_student_marks(marks_list):
```

```
        return sorted(marks_list, reverse=True)

    student_marks = [85, 92, 75, 92, 88, 70, 85]
    sorted_marks = sort_student_marks(student_marks)
    print("Sorted student marks:", sorted_marks)
```

```
    Sorted student marks: [92, 92, 88, 85, 85, 75, 70]
```

Task-3:Prime Number Validation

Few-Shot Prompting for Prime Number Validation: Provide multiple input-output examples for a function that checks whether a number is prime. Observe how few-shot prompting improves correctness.

```python
def is_prime(n):
    if n <= 1:
        return False
    for i in range(2, int(n**0.5) + 1):
        if n % i == 0:
            return False
    return True
# Test cases
print(is_prime(2))
print(is_prime(4))
print(is_prime(17))
print(is_prime(1))
print(is_prime(19))
print(is_prime(20))
print(is_prime(23))
print(is_prime(25))
print(is_prime(29))
print(is_prime(30))
print(is_prime(97))
print(is_prime(100))
print(is_prime(121))
print(is_prime(131))
```

```
    True
    False
    True
    False
    True
    False
    True
    False
    True
    False
    True
    False
    False
    True
```

Task-4:Student Grading System (UI)

Prompt-Guided UI Design for Student Grading System: Create a user interface for a student grading system that calculates total marks, percentage, and grade based on user input.

```python
def calculate_total_and_percentage(marks):
    total = sum(marks)
    percentage = (total / (len(marks) * 100)) * 100  # Assuming each subject is out of 100
    return total, percentage
marks = []
num_subjects = int(input("Enter number of subjects: "))
for i in range(num_subjects):
    mark = float(input(f"Enter mark for subject {i+1}: "))
    marks.append(mark)
total, percentage = calculate_total_and_percentage(marks)
print(f"Total marks: {total}")
print(f"Percentage: {percentage:.2f}%")
```

```
    Enter number of subjects: 3
    Enter mark for subject 1: 19
    Enter mark for subject 2: 20
    Enter mark for subject 3: 15
    Total marks: 54.0
    Percentage: 18.00%
```

Task-5:Unit Conversion

Analyzing Prompt Specificity in Unit Conversion Functions: Improving a Unit Conversion Function (Kilometers to Miles and Miles to Kilometers) Using Clear Instructions.

```python
def km_to_miles(km):
    miles = km * 0.621371
    return miles
# Test the function
kilometers = float(input("Enter distance in kilometers: "))
miles = km_to_miles(kilometers)
print(f"{kilometers} kilometers is equal to {miles} miles.")
```

```
Enter distance in kilometers: 21
21.0 kilometers is equal to 13.048791 miles.
```