

Python Program

```
def main():  
    print('Hello, World! My name is Praveen')  
  
if __name__ == '__main__':  
    main()
```

Sample Input & Expected Output

Input: (none)

Output: Hello, World! My name is Praveen

Python Program

```
def char_frequency(s):  
    freq = {}  
    for ch in s:  
        freq[ch] = freq.get(ch, 0) + 1  
    return freq  
  
print(char_frequency('nandini'))
```

Sample Input & Expected Output

Input: 'nandini'

Output: {'n':3,'a':1,'d':1,'i':2}

Python Program

```
def pascal_triangle(n):  
    triangle = []  
    for i in range(n):  
        row = [1]*(i+1)  
        for j in range(1,i):  
            row[j] = triangle[i-1][j-1] + triangle[i-1][j]  
        triangle.append(row)  
    return triangle  
  
print(pascal_triangle(5))
```

Sample Input & Expected Output

Input: 5

Output: [[1],[1,1],[1,2,1],[1,3,3,1],[1,4,6,4,1]]

Python Program

```
def most_frequent_word(words):  
    freq = {}  
    for w in words:  
        freq[w] = freq.get(w, 0) + 1  
    return max(freq, key=freq.get)  
  
print(most_frequent_word(['apple','banana','apple','Venkatesh','banana','apple']))
```

Sample Input & Expected Output

Input: ['apple','banana','apple','Venkatesh','banana','apple']

Output: 'apple'

Python Program

```
class BankAccount:
    def __init__(self, name, balance=0):
        self.name = name
        self.balance = balance
    def deposit(self, amount):
        self.balance += amount
    def withdraw(self, amount):
        if amount <= self.balance:
            self.balance -= amount
        else:
            raise ValueError('Insufficient funds')
    def get_balance(self):
        return self.balance

acc = BankAccount('Srawani', 1000)
acc.deposit(500)
acc.withdraw(300)
print(acc.get_balance())
```

Sample Input & Expected Output

Input: Deposit 200, Withdraw 100

Output: 1100

Python Program

```
num = int(input("Enter a number: "))

factorial = 1

if num < 0:

    print(" Factorial does not exist for negative numbers")

elif num == 0:

    print("The factorial of 0 is 1")

else:

    for i in range(1,num + 1):

        factorial = factorial*i

    print ("The factorial of", num,"is", factorial)
```

Sample Input & Expected Output

Enter a number: -1

Factorial does not exist for negative numbers

Enter a number: 9

The factorial of 9 is 362880

Source Code:

```
num1 = float(input("Enter the first number: "))
num2 = float(input("Enter the second number: "))
num3 = float(input("Enter the third number: "))

# Step 2: Compare the numbers to find the largest

if num1 > num2 and num1 > num3:
    largest = num1
elif num2 > num3:
    largest = num2
else:
    largest = num3

print(f"The largest number is {largest}")
```

Sample Input & Expected Output

Output:

Enter the first number: 85

Enter the second number: 74

Enter the third number: 112

The largest number is 112.0

Source Code:

```
lower = int(input("enter lower range:"))
upper = int(input("enter upper range:"))
print("Prime numbers between", lower, "and", upper, "are:")
for num in range(lower, upper + 1):
    # all prime numbers are greater than 1
    if num > 1:
        for i in range(2, num):
            if (num % i) == 0:
                break
        else:
            print(num)
```

Sample Input & Expected Output

enter lower range:26

enter upper range:42

Prime numbers between 26 and 42 are:

29

31

37

41

Source Code:

```
num1 = int(input("Enter first number: "))
num2 = int(input("Enter second number: "))

# Find the greater number

if num1 > num2:
    lcm = num1
else:
    lcm = num2

# Loop until LCM is found

while True:
    if lcm % num1 == 0 and lcm % num2 == 0:
        break
    lcm += 1

# Print the result

print(f"The LCM of {num1} and {num2} is {lcm}")
```

Sample Input & Expected Output

Enter first number: 35

Enter second number: 55

The LCM of 35 and 55 is 385

Source Code:

```
n = int(input("Enter the number of terms in the Fibonacci sequence: "))

# Initialize the first two Fibonacci numbers
a, b = 0, 1

print("Fibonacci sequence:")

# Loop to print the first n terms
for i in range(n):
    print(a, end=" ")
    # Calculate the next term
    next_term = a + b
    a = b
    b = next_term
```

Sample Input & Expected Output

Enter the number of terms in the Fibonacci sequence: 18

Fibonacci sequence:

0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597

Source Code:

```
def sum_natural_numbers(num):  
    # Using the formula: sum = n(n+1)/2  
    return num * (num + 1) // 2  
  
def main():  
    try:  
        # Ask the user for input  
        num = int(input("Enter a positive integer: "))  
        if num <= 0:  
            print("Please enter a number greater than 0.")  
        else:  
            total = sum_natural_numbers(num)  
            print(f"Sum of natural numbers up to {num} is {total}")  
    except ValueError:  
        print("Invalid input. Please enter a valid integer.")  
  
# Run the program  
if __name__ == "__main__":  
    main()
```

Sample Input & Expected Output

Enter a positive integer: 15

Sum of natural numbers up to 15 is 120

Source Code:

```
# Python Calculator Program

# Function to add two numbers

def add(x, y):

    return x + y

# Function to subtract two numbers

def subtract(x, y):

    return x - y

# Function to multiply two numbers

def multiply(x, y):

    return x * y

# Function to divide two numbers

def divide(x, y):

    if y == 0:

        return "Error! Division by zero."

    return x / y

# Display menu

def display_menu():

    print("\nSelect operation:")

    print("1. Add")

    print("2. Subtract")

    print("3. Multiply")

    print("4. Divide")

    print("5. Exit")

# Main program loop
```

```
while True:

    display_menu()

    choice = input("Enter choice (1/2/3/4/5): ")

    if choice == '5':

        print("Exiting the calculator. Thank you!")

        break

    if choice in ('1', '2', '3', '4'):

        try:

            num1 = float(input("Enter first number: "))

            num2 = float(input("Enter second number: "))

        except ValueError:

            print("Invalid input! Please enter numeric values.")

            continue

        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)}")

    else:

        print("Invalid choice! Please select a valid option.")
```

Sample Input & Expected Output

Select operation:

1. Add
2. Subtract
3. Multiply
4. Divide
5. Exit

Enter choice (1/2/3/4/5): 1

Enter first number: 5

Enter second number: 6

$5.0 + 6.0 = 11.0$

Select operation:

1. Add
2. Subtract
3. Multiply
4. Divide
5. Exit

Enter choice (1/2/3/4/5): 2

Enter first number: 6

Enter second number: 7

$6.0 - 7.0 = -1.0$

Select operation:

1. Add
2. Subtract
3. Multiply
4. Divide

5. Exit

Enter choice (1/2/3/4/5): 3

Enter first number: 56

Enter second number: 87

$56.0 * 87.0 = 4872.0$

Select operation:

1. Add

2. Subtract

3. Multiply

4. Divide

5. Exit

Enter choice (1/2/3/4/5): 5

Exiting the calculator. Thank you!

Source Code:

```
# Program to check if a string is a palindrome

# Read string input from user
string = input("Enter a string: ")

# Remove spaces and convert to lowercase for uniformity
clean_string = string.replace(" ", "").lower()

# Reverse the string
reversed_string = clean_string[::-1]

# Check if original and reversed strings are the same
if clean_string == reversed_string:
    print(f"{string}" is a palindrome.')
else:
    print(f"{string}" is not a palindrome.')
```

Sample Input & Expected Output

Enter a string: madam

"madam" is a palindrome.

Enter a string: lalitha

"lalitha" is not a palindrome.

Source Code:

```
# Program to add two matrices using nested loops

# Read number of rows and columns

rows = int(input("Enter the number of rows: "))

cols = int(input("Enter the number of columns: "))

# Initialize the matrices

matrix1 = []

matrix2 = []

# Input elements of the first matrix

print("Enter elements of the first matrix:")

for i in range(rows):

    row = []

    for j in range(cols):

        num = int(input(f"Element [{i+1},{j+1}]: "))

        row.append(num)

    matrix1.append(row)

# Input elements of the second matrix

print("Enter elements of the second matrix:")

for i in range(rows):

    row = []

    for j in range(cols):

        num = int(input(f"Element [{i+1},{j+1}]: "))

        row.append(num)

    matrix2.append(row)

# Add the two matrices

result = []
```

```
for i in range(rows):  
    row = []  
    for j in range(cols):  
        row.append(matrix1[i][j] + matrix2[i][j])  
    result.append(row)  
# Display the result  
print("\nSum of the two matrices:")  
for row in result:  
    print(row)
```

Sample Input & Expected Output

Enter the number of rows: 2

Enter the number of columns: 2

Enter elements of the first matrix:

Element [1,1]: 1

Element [1,2]: 3

Element [2,1]: 5

Element [2,2]: 6

Enter elements of the second matrix:

Element [1,1]: 5

Element [1,2]: 6

Element [2,1]: 7

Element [2,2]: 0

Sum of the two matrices:

[6, 9]

[12, 6]

Source Code:

```
# Program to display all Armstrong numbers in a given interval

# Input the interval

lower = int(input("Enter the lower bound: "))
upper = int(input("Enter the upper bound: "))

print(f"Armstrong numbers between {lower} and {upper} are:")

# Loop through all numbers in the interval
for num in range(lower, upper + 1):
    num_digits = len(str(num))

    sum_of_powers = 0

    temp = num

    # Calculate sum of digits raised to the power of num_digits
    while temp > 0:
        digit = temp % 10

        sum_of_powers += digit ** num_digits

        temp = temp // 10

    # Check if sum equals the original number
    if sum_of_powers == num:
        print(num)
```

Sample Input & Expected Output

Enter the lower bound: 10

Enter the upper bound: 500

Armstrong numbers between 10 and 500 are:

153

370

371

407