Program -1

```
s = input("Enter a string: ").lower()

vowels = 0
consonants = 0

for ch in s:
    if ch.isalpha():
        if ch in 'aeiou':
            vowels += 1
        else:
            consonants += 1

print("Number of vowels:", vowels)
print("Number of consonants:", consonants)
```

```
========== RESTART: C:\Users\glaks\OneDrive\Desktop\sem4\ml\program 01.py ==========
Enter a string: hi how are u this is lakshmi pranathi
Number of vowels: 12
Number of consonants: 18
```

Program2

```python
def read_matrix(rows, cols, name):
    matrix = []
    for i in range(rows):
        row = list(map(int, input(f"Row {i+1}: ").split()))
        matrix.append(row)
    return matrix
def multiply_matrices(A, B):
    result = [[0 for _ in range(len(B[0]))] for _ in range(len(A))]

    for i in range(len(A)):
        for j in range(len(B[0])):
            for k in range(len(B)):
                result[i][j] += A[i][k] * B[k][j]
    return result
r1, c1 = map(int, input("Enter rows and columns of Matrix A: ").split())
r2, c2 = map(int, input("Enter rows and columns of Matrix B: ").split())
if c1 != r2:
    print("\nError: Matrix multiplication not possible!")
else:
    A = read_matrix(r1, c1, "A")
    B = read_matrix(r2, c2, "B")

    product = multiply_matrices(A, B)

    print("\nProduct Matrix (A × B):")
    for row in product:
        print(row)
```

```
========== RESTART: C:/Users/glaks/OneDrive/Desktop/sem4/ml/program02.py ==========
Enter rows and columns of Matrix A:  3 3
Enter rows and columns of Matrix B:  3 3
Row 1: 3 4 5
Row 2: 3 4 5
Row 3: 3 4 5
Row 1: 1 2 3
Row 2: 1 2  3
Row 3: 1 2 3

Product Matrix (A × B):
[12, 24, 36]
[12, 24, 36]
[12, 24, 36]
```

Program-3

```python
list1 = list(map(int, input("Enter elements of first list: ").split()))
list2 = list(map(int, input("Enter elements of second list: ").split()))
common_elements = set(list1) & set(list2)
print("Number of common elements:", len(common_elements))
print("Common elements:", list(common elements))
```

```
========== RESTART: C:/Users/glaks/OneDrive/Desktop/sem4/ml/program03.py =========
Enter elements of first list: 3 4 5
Enter elements of second list: 4 5 6
Number of common elements: 2
Common elements: [4, 5]
```

Program-4

```
rows, cols = map(int, input("Enter number of rows and columns: ").split())
matrix = []
print("Enter matrix elements:")
for i in range(rows):
    row = list(map(int, input(f"Row {i+1}: ").split()))
    matrix.append(row)
transpose = [[matrix[j][i] for j in range(rows)] for i in range(cols)]
for row in transpose:
    print(row)
```

```
========== RESTART: C:/Users/glaks/OneDrive/Desk
Enter number of rows and columns: 4 4
Enter matrix elements:
Row 1: 3 4 5 6
Row 2: 2 3 4 5
Row 3: 5 6 7 8
Row 4: 5 6 8 9
[3, 2, 5, 5]
[4, 3, 6, 6]
[5, 4, 7, 8]
[6, 5, 8, 9]
```

Program-5

program05.py - C:/Users/glaks/OneDrive/Desktop/sem4/ml/program05.py (3.10.6)

File Edit Format Run Options Window Help

```
numbers = [(100 + (i % 51)) for i in range(100)]
total = 0
for num in numbers:
    total += num
mean = total / len(numbers)
numbers.sort()
n = len(numbers)
if n % 2 == 0:
    median = (numbers[n//2 - 1] + numbers[n//2]) / 2
else:
    median = numbers[n//2]
max_count = 0
mode = numbers[0]

for i in numbers:
    count = 0
    for j in numbers:
        if j == i:
            count += 1
    if count > max_count:
        max_count = count
        mode = i
print("Generated Numbers:")
print(numbers)

print("\nMean:", mean)
print("Median:", median)
print("Mode:", mode)
```

```
========== RESTART: C:/Users/glaks/OneDrive/Desktop/sem4/ml/program05.py ==========
Generated Numbers:
[100, 100, 101, 101, 102, 102, 103, 103, 104, 104, 105, 105, 106, 106, 107, 107, 108, 108, 109, 109, 110, 110, 111, 111, 112, 112, 113, 113, 114, 114, 115, 115, 116, 116, 117, 117, 118
, 118, 119, 119, 120, 120, 121, 121, 122, 122, 123, 123, 124, 124, 125, 125, 126, 126, 127, 127, 128, 128, 129, 129, 130, 130, 131, 131, 132, 132, 133, 133, 134, 134, 135, 135, 136, 13
6, 137, 137, 138, 138, 139, 139, 140, 140, 141, 141, 142, 142, 143, 143, 144, 144, 145, 145, 146, 146, 147, 147, 148, 148, 149, 150]

Mean: 124.51
Median: 124.5
Mode: 100
```