**Task-1 a :**

**Program:**

# Get input values

a = int(input("Enter value for a: "))

b = int(input("Enter value for b: "))

# Arithmetic Expressions

print("Arithmetic Expressions:")

print("a + b =", a + b)

print("a - b =", a - b)

print("a \* b =", a \* b)

print("a / b =", a / b)

print("a % b =", a % b)

print("a // b =", a // b)

print("a \*\* b =", a \*\* b)

# Relational Expressions

print("\nRelational Expressions:")

print("a > b:", a > b)

print("a < b:", a < b)

print("a == b:", a == b)

print("a != b:", a != b)

print("a >= b:", a >= b)

print("a <= b:", a <= b)

# Logical Expressions

print("\nLogical Expressions:")

print("a > 0 and b > 0:", a > 0 and b > 0)

print("a > 0 or b > 0:", a > 0 or b > 0)

print("not(a > 0):", not(a > 0))

# Assignment Expressions

print("\nAssignment Expressions:")

x = a

print("Initial x =", x)

x += b

print("x += b:", x)

x -= b

print("x -= b:", x)

x \*= b

print("x \*= b:", x)

x /= b

print("x /= b:", x)

**Sample Input/Output:**

Enter value for a: 10

Enter value for b: 5

Arithmetic Expressions:

a + b = 15

a - b = 5

a \* b = 50

a / b = 2.0

a % b = 0

a // b = 2

a \*\* b = 100000

Relational Expressions:

a > b: True

a < b: False

a == b: False

a != b: True

a >= b: True

a <= b: False

Logical Expressions:

a > 0 and b > 0: True

a > 0 or b > 0: True

not(a > 0): False

Assignment Expressions:

Initial x = 10

x += b: 15

x -= b: 10

x \*= b: 50

x /= b: 10.0

**task-1 b:**

**Program:**

# Input temperature in Celsius

celsius = float(input("Enter temperature in Celsius: "))

# Convert to Fahrenheit

fahrenheit = (celsius \* 9/5) + 32

# Display the result

print(f"{celsius} Celsius is equal to {fahrenheit} Fahrenheit.")

**Sample Input/Output:**

Enter temperature in Celsius: 30

30.0 Celsius is equal to 86.0 Fahrenheit.

**Task -1 c :**

**Program :**

import cmath # To handle both real and complex roots

# Input coefficients

a = float(input("Enter coefficient a: "))

b = float(input("Enter coefficient b: "))

c = float(input("Enter coefficient c: "))

# Calculate discriminant

D = b\*\*2 - 4\*a\*c

# Compute roots using quadratic formula

root1 = (-b + cmath.sqrt(D)) / (2 \* a)

root2 = (-b - cmath.sqrt(D)) / (2 \* a)

# Display results

print(f"\nDiscriminant (D) = {D}")

# Check nature of roots

if D > 0:

print("The roots are real and distinct.")

elif D == 0:

print("The roots are real and equal.")

else:

print("The roots are complex.")

# Display the roots

print(f"Root 1 = {root1}")

print(f"Root 2 = {root2}")

**Sample Input/Output**

a. Enter coefficient a: 1

Enter coefficient b: -4

Enter coefficient c: 4

Discriminant (D) = 0.0

The roots are real and equal.

Root 1 = (2+0j)

Root 2 = (2+0j)

b. Enter coefficient a: 1

Enter coefficient b: -5

Enter coefficient c: 6

Discriminant (D) = 1.0

The roots are real and distinct.

Root 1 = (3+0j)

Root 2 = (2+0j)

c. Enter coefficient a: 1

Enter coefficient b: 2

Enter coefficient c: 5

Discriminant (D) = -16.0

The roots are complex.

Root 1 = (-1+2j)

Root 2 = (-1-2j)

**Task -1 d :**

**Program**

X = float(input("Enter the cost price of the scooter (Rs.): "))

Y = float(input("Enter the repair cost (Rs.): "))

Z = float(input("Enter the selling price (Rs.): "))

# Calculate total cost price

cost\_price = X + Y

# Ensure selling price is greater than cost price

if Z <= cost\_price:

print("No gain. Selling price must be greater than total cost.")

else:

gain = Z - cost\_price

gain\_percent = (gain / cost\_price) \* 100

# Display results

print(f"Total Cost Price = Rs. {cost\_price}")

print(f"Gain = Rs. {gain}")

print(f"Gain Percent = {gain\_percent:.2f}%")

Sample Input/Output

Enter the cost price of the scooter (Rs.): 50000

Enter the repair cost (Rs.): 3000

Enter the selling price (Rs.): 58000

Total Cost Price = Rs. 53000.0

Gain = Rs. 5000.0

Gain Percent = 9.43%

**Task -1 e :**

**Program :**

total\_systems = int(input("Enter the total number of systems in the lab: "))

# Percent distribution

dell\_percent = 36

lenovo\_percent = 34

acer\_percent = 28

samsung\_percent = 2

# Count per brand using integer rounding

dell\_count = total\_systems \* dell\_percent // 100

lenovo\_count = total\_systems \* lenovo\_percent // 100

acer\_count = total\_systems \* acer\_percent // 100

samsung\_count = total\_systems \* samsung\_percent // 100

# Print results using sep

print("Total Systems:", total\_systems)

print("Dell:", dell\_count, sep="\t")

print("Lenovo:", lenovo\_count, sep="\t")

print("Acer:", acer\_count, sep="\t")

print("Samsung:", samsung\_count, sep="\t")

or

print("Total Systems:", total\_systems)

print("Dell:", dell\_count, "\nLenovo:", lenovo\_count, "\nAcer:", acer\_count, "\nSamsung:", samsung\_count, sep="\t")

Sample input/output

Enter the total number of systems in the lab: 150

Total Systems: 150

Dell: 54

Lenovo: 51

Acer:42

Samsung: 3