1. Problem Description:

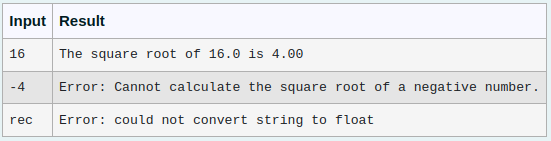
Develop a Python program that safely calculates the square root of a number provided by the user. Handle exceptions for negative inputs and non-numeric inputs.

Input Format:

User inputs a number.

Output Format:

Print the square root of the number or an error message if an exception occurs.



import math

try:

a=float(input())

if(a>0):

b=math.sqrt(a)

print(f"The square root of {a} is {b:.2f}")

elif(a==0):

b=math.sqrt(a)

print(f"The square root of {a} is {b:.2f}")

elif(a<0):

print("Error: Cannot calculate the square root of a negative number.")

except ValueError:

print("Error: could not convert string to float")

2.Problem Description:

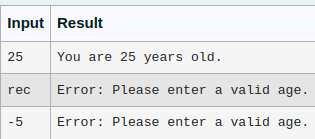
Write a Python program that asks the user for their age and prints a message based on the age. Ensure that the program handles cases where the input is not a valid integer.

Input Format:

A single line input representing the user's age.

Output Format:

Print a message based on the age or an error if the input is invalid



try:

age = int(input(""))

if age < 0:

print("Error: Please enter a valid age.")

else:

print(f"You are {age} years old.")

except ValueError:

print("Error: Please enter a valid age.")

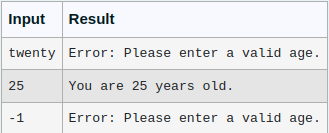
except EOFError:

print("Error: Please enter a valid age.")

3. Write a Python program that asks the user for their age and prints a message based on the age. Ensure that the program handles cases where the input is not a valid integer.

**Input Format:** A single line input representing the user's age.

**Output Format:** Print a message based on the age or an error if the input is invalid.



try:

age = int(input(""))

if age < 0:

print("Error: Please enter a valid age.")

else:

print(f"You are {age} years old.")

except ValueError:

print("Error: Please enter a valid age.")

except EOFError:

print("Error: Please enter a valid age.")

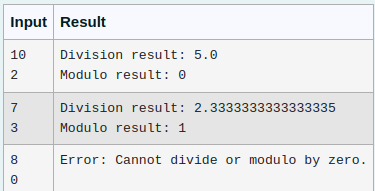
4. Write a Python program that performs division and modulo operations on two numbers provided by the user. Handle division by zero and non-numeric inputs.

Input Format:

Two lines of input, each containing a number.

Output Format:

Print the result of division and modulo operation, or an error message if an exception occurs.



try:

num1 = float(input(""))

num2 = float(input(""))

division\_result = num1 / num2

modulo\_result = num1 % num2

print(f"Division result: {division\_result:}")

print(f"Modulo result: {modulo\_result:.0f}")

except ZeroDivisionError:

print("Error: Cannot divide or modulo by zero.")

except ValueError:

print("Error: Non-numeric input provided.")

5. Problem Description:

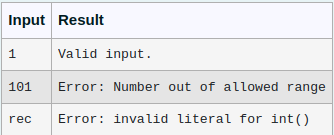
Write a Python script that asks the user to enter a number within a specified range (e.g., 1 to 100). Handle exceptions for invalid inputs and out-of-range numbers.

Input Format:

User inputs a number.

Output Format:

Confirm the input or print an error message if it's invalid or out of range.



try:

lower\_limit = 1

upper\_limit = 100

num = int(input(""))

if num < lower\_limit or num > upper\_limit:

print("Error: Number out of allowed range")

else:

print("Valid input.")

except ValueError:

print("Error: invalid literal for int()")