

## 11 – Exceptions

Ex. No.: 11.1

Date:

Register No.: 2116231501105

Name: Nandhini Prakash

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.

**For example:**

Input	Result
10 2	Division result: 5.0 Modulo result: 0
7 3	Division result: 2.3333333333333335 Modulo result: 1
8 0	Error: Cannot divide or modulo by zero.

```
1 try:
2     n = int(input())
3     m = int(input())
4     print("Division result:",n/m)
5     print("Modulo result:",n%m)
6 except ZeroDivisionError:
7     print("Error: Cannot divide or modulo by zero.")
8 except ValueError:
9     print("Error: Non-numeric input provided.")
10
```

Output:

✓	8 0	Error: Cannot divide or modulo by zero.	Error: Cannot divide or modulo
✓	abc 5	Error: Non-numeric input provided.	Error: Non-numeric input provid

Passed all tests! ✓

Ex. No.: 11.2

Date:

Register No.: 2116231501105

Name: Nandhini Prakash

Develop a Python program that safely performs division between two numbers provided by the user. Handle exceptions like division by zero and non-numeric inputs.

**Input Format:** Two lines of input, each containing a number.

**Output Format:** Print the result of the division or an error message if an exception occurs.

**For example:**

Input	Result
10 2	5.0
10 0	Error: Cannot divide or modulo by zero.
ten 5	Error: Non-numeric input provided.

```
1 try:
2     n = float(input())
3     m = float(input())
4     print(n/m)
5 except ZeroDivisionError:
6     print("Error: Cannot divide or modulo by zero.")
7 except ValueError:
8     print("Error: Non-numeric input provided.")
9
```

Output:

	Input	Expected	Got
✓	10 2	5.0	5.0
✓	10 0	Error: Cannot divide or modulo by zero.	Error: Cannot divide or mo
✓	ten 5	Error: Non-numeric input provided.	Error: Non-numeric input p

Passed all tests! ✓

Ex. No.: 11.3

Date:

Register No.: 2116231501105

Name: Nandhini Prakash

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.

**For example:**

Input	Result
twenty	Error: Please enter a valid age.
25	You are 25 years old.
-1	Error: Please enter a valid age.

**Answer:** (penalty regime: 0 %)

```
1 try:
2     n = int(input())
3     if(n<0):
4         print("Error: Please enter a valid age.")
5     else:
6         print("You are",n,"years old.")
7 except:
8     print("Error: Please enter a valid age.")
```

**Output:**

	Input	Expected	Got	
✓	twenty	Error: Please enter a valid age.	Error: Please enter a valid age.	✓
✓	25	You are 25 years old.	You are 25 years old.	✓
✓	-1	Error: Please enter a valid age.	Error: Please enter a valid age.	✓
✓	150	You are 150 years old.	You are 150 years old.	✓
✓		Error: Please enter a valid age.	Error: Please enter a valid age.	✓

Passed all tests! ✓

**Correct**

Marks for this submission: 1.00/1.00.

Ex. No.: 11.4

Date:

Register No.: 2116231501105

Name: Nandhini Prakash

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.

**For example:**

Input	Result
1	Valid input.
101	Error: Number out of allowed range
rec	Error: invalid literal for int()

**Answer:** (penalty regime: 0 %)

```
1 try:
2     n = int(input())
3     if(n>100 or n<1):
4         print("Error: Number out of allowed range")
5     else:
6         print("Valid input.")
7 except:
8     print("Error: invalid literal for int()")
```

Output:

	Input	Expected	Got
✓	1	Valid input.	Valid input.
✓	100	Valid input.	Valid input.
✓	101	Error: Number out of allowed range	Error: Number out of allowed range

Passed all tests! ✓

Ex. No.: 11.5

Date:

Register No.: 2116231501105

Name: Nandhini Prakash

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.

**For example:**

Input	Result
16	The square root of 16.0 is 4.00
-4	Error: Cannot calculate the square root of a negative number.
rec	Error: could not convert string to float

**Answer:** (penalty regime: 0 %)

```
1 try:
2     n = float(input())
3     if(n<0):
4         print("Error: Cannot calculate the square root of a negative number.")
5     else:
6         print("The square root of",n,"is", "%.2f"%(n**0.5))
7 except:
8     print("Error: could not convert string to float")
```

Output:

	Input	Expected	Got
✓	16	The square root of 16.0 is 4.00	The square root of 16.0 is 4.00
✓	0	The square root of 0.0 is 0.00	The square root of 0.0 is 0.00
✓	-4	Error: Cannot calculate the square root of a negative number.	Error: Cannot calculate the square root of a negative number.

Passed all tests! ✓

**Correct**

Marks for this submission: 1.00/1.00.