11-EXCEPTIONHANDLING

EXCEPTIONHANDLING

To find whether a digit lies in the specified range(1-100). Handling exceptions forinvalid inputs and out-of-range numbers .

InputFormat:

Userinputsanumber. Output

Format:

Confirmtheinputorprintanerrormessageifit'sinvalidoroutofrange.

Forexample:

Input	Result
1	Validinput.
101	Error:Numberoutofallowedrange
rec	Error:invalidliteralforint()

Program:

```
try:

   a=input()

   if(int(a)>0andint(a)<101):

      print("Valid input.")

   else:

      print("Error:Numberoutofallowedrange") except:

   print("Error:invalidliteralforint()")
```

	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	✔

EXCEPTIONHANDLING

Write a Python program that performs division and modulo operations on two numbers providedbytheuser. Handledivisionbyzeroandnon-numericinputs.

InputFormat:

Twolinesofinput,eachcontaininganumber.

Output Format:

Printtheresultofdivisionandmodulooperation,oranerrormessageifan exception occurs.

Forexample:

Input	Result
10 2	Divisionresult:5.0 Moduloresult:0
7 3	Divisionresult:2.3333333333333335 Moduloresult:1
8 0	Error:Cannotdivideormodulobyzero.

Program:

```
try:

    a=input()

    b=input()

    c=int(a)/int(b)

    d=int(a)%int(b)

exceptZeroDivisionError:

    print("Error:Cannotdivideormodulobyzero.") except:

    print("Error:Non-numericinputprovided.") else:

    print("Divisionresult:",c)

    print("Moduloresult:",d)
```

	Input	Expected	Got
✔	10 2	Division result: 5.0 Modulo result: 0	Division result: 5.0 Modulo result: 0
✔	7 3	Division result: 2.3333333333333335 Modulo result: 1	Division result: 2.3333333333333335 Modulo result: 1
✔	8 0	Error: Cannot divide or modulo by zero.	Error: Cannot divide or modulo by zero.
✔	abc 5	Error: Non-numeric input provided.	Error: Non-numeric input provided.

EXCEPTIONHANDLING

WriteaPythonprogramthataskstheuserfortheirageandprintsamessagebasedon the age. Ensure that the program handles cases where the input is not a valid integer.

InputFormat: Asingle lineinputrepresentingtheuser's age.

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

Forexample:

Input	Result
twenty	Error:Pleaseenteravalid age.
25	Youare25yearsold.
-1	Error:Pleaseenteravalid age.

Program:

```
try:
    a=input()
    ifint(a)>=0:
        print("Youare",a,"yearsold.") else:
        print("Error:Pleaseenteravalidage.") except:
    print("Error:Pleaseenteravalidage.")
```

	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.	✔

Ex.No.   :   11.4   **Date:02.06.24**

RegisterNo.:230701369   **Name:VALLURU VARSHINI**

EXCEPTIONHANDLING

DevelopaPythonprogramthatsafelycalculatesthesquarerootofanumber provided by the user. Handle exceptions for negative inputs and non-numeric inputs.

InputFormat:

Userinputsanumber. Output

Format:

Printthesquarerootofthenumberoranerrormessageifanexceptionoccurs.

Forexample:

Input	Result
16	Thesquarerootof16.0is4.00
-4	Error:Cannotcalculatethesquarerootofanegativenumber.
rec	Error:couldnotconvertstringtofloat

Program:

```
importmath
try:
    n=input()
    n=float(n)
    if n < 0:
        print("Error:Cannotcalculatethesquarerootofanegativenumber.") else:
        r=math.sqrt(n)
```

```
    print("Thesquarerootof{}is{:.2f}".format(n,r))


exceptValueError:

    print("Error:couldnotconvertstringtofloat")
```

	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.	✔

EXCEPTIONHANDLING

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.

InputFormat: Twolinesofinput,eachcontaininganumber.

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

Forexample:

Input	Result
10 2	5.0
10 0	Error:Cannotdivideormodulobyzero.
ten 5	Error:Non-numericinputprovided.

Program:

```
try:
    a=input()
    b=input()
    c=float(a)/float(b)
exceptZeroDivisionError:
```

print("Error:Cannotdivideormodulobyzero.") except:

print("Error:Non-numericinputprovided.") else:

print(c)

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