11. EXCEPTION HANDLING

Ex. No. : 11.1 Date: 02.06.24

Register No.: 2307010105 Name: Harish A

EXCEPTION HANDLING

Ľo find whetheí a digit lies in the specified íange(1-100). Handling exceptions foí invalidinputs and out-of-íange numbeís .

Input Foímat:

Useí inputs a numbeí.

Output Foímat:

Confiím the input oí píint an eíioí message if it's invalid oí out of íange.

For example:

Input	Result
1	Valid input.
101	Eíioí: Numbeí out of allowed íange
íec	Eíioí: invalid liteíal foí int()

Program:

```
try:
    a=input()
    if(int(a)>0 and int(a)<101):
        print("Valid input.")
    else:
        print("Error: Number out of allowed range")
except:
```

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

Ex. No. : 11.2 Date: 02.06.24

Register No.: 230701105 Name Harish A

EXCEPTION HANDLING

Wíite a Python píogíam that peífoíms division and modulo opeíations on two numbeís píovided by the useí. Handle division by zeío and non-numeíic inputs.

Input Foímat:

L'wo lines of input, each containing a numbeí.

Output Foímat:

Píint the íesult of division and modulo opeíation, oí an eííoí message if an exception occuís.

For example:

Input	Result
10 2	Division íesult: 5.0 Modulo íesult: 0
7 3	Division íesult: 2.3333333333333335 Modulo íesult: 1

Input	Result
8 0	Eíío: Cannot divide oí modulo by zeío.

Program:

```
try:
    a=input()
    b=input()
    c=int(a)/int(b)
    d=int(a)%int(b)
except ZeroDivisionError:
    print("Error: Cannot divide or modulo by zero.")
except:
    print("Error: Non-numeric input provided.")
else:
    print("Division result:",c)
    print("Modulo result:",d)
```

Ex. No. : 11.3 Date: 02.06.24

Register No.: 230701105 Name: Harish A

EXCEPTION HANDLING

Wíite a Python píogíam that asks the useí foí theií age and píints a message based on the age. Ensuíe that the píogíam handles cases wheíe the input is not a valid integeí.

Input Format: A single line input íepíesenting the useí's age.

Output Format: Píint a message based on the age oí an eíioí if the input is invalid.

For example:

Input	Result
twenty	Eíioí: Please enteí a validage.

Input	Result
25	You aíe 25 yeaís old.
-1	Eíioí: Please enteí a validage.

Program:

```
try:
    a=input()
    if int(a)>=0:
        print("You are",a,"years old.")
    else:
        print("Error: Please enter a valid age.")
except:
    print("Error: Please enter a valid age.")
```

Ex. No. : 11.4 Date: 02.06.24

Register No.: 230701105 Name: Harish A

EXCEPTION HANDLING

Develop a Python píogíam that safely calculates the squaíe íoot of a numbeí píovidedby the useí. Handle exceptions foí negative inputs and non-numeíic inputs.

Input Foímat:

Useí inputs a numbeí.

Output Foímat:

Píint the squaíe íoot of the numbeí oí an eíioí message if an exception occuís.

For example:

Input	Result
16	Ľhe squaíe íoot of 16.0 is 4.00
-4	Eíío í: Cannot calculate the squaíe íoot of a negative numbeí.
íec	Eíío í: could not conveít stíing to float

Program:

```
import math
try:
   n=input()
   n=float(n)
   if n < 0:
      print("Error: Cannot calculate the square root of a negative number.")
   else:
      r= math.sqrt(n)
      print("The square root of {} is {:.2f}".format(n, r))except ValueError:
   print("Error: could not convert string to float")
```

Ex. No. : 11.5 Date:

Register No.: 230701105 Name: Harish A

EXCEPTION HANDLING

Develop a Python píogíam that safely peífoíms division between two numbeís píovided by the useí. Handle exceptions like division by zeío and non-numeíic inputs.

Input Format: Ľwo lines of input, each containing a numbeí.

Output Format: Píint the íesult of the division oí an eíío í message if an exceptionoccuís.

For example:

Input	Result
10 2	5.0
10 0	Eííoí: Cannot divide oí modulo by zeío.
ten 5	Eííoí: Non-numeíic input píovided.

Program:

```
try:
    a=input()
    b=input()
    c=float(a)/float(b)
except ZeroDivisionError:
    print("Error: Cannot divide or modulo by zero.")
except:
    print("Error: Non-numeric input provided.")
else:
    print(c)
```