011 – EXCEPTIONS

Ex. No. : 11.1 **Date: 6-6-24**

Register No.: 231501140 **Name: Sai Senthil .M**

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.

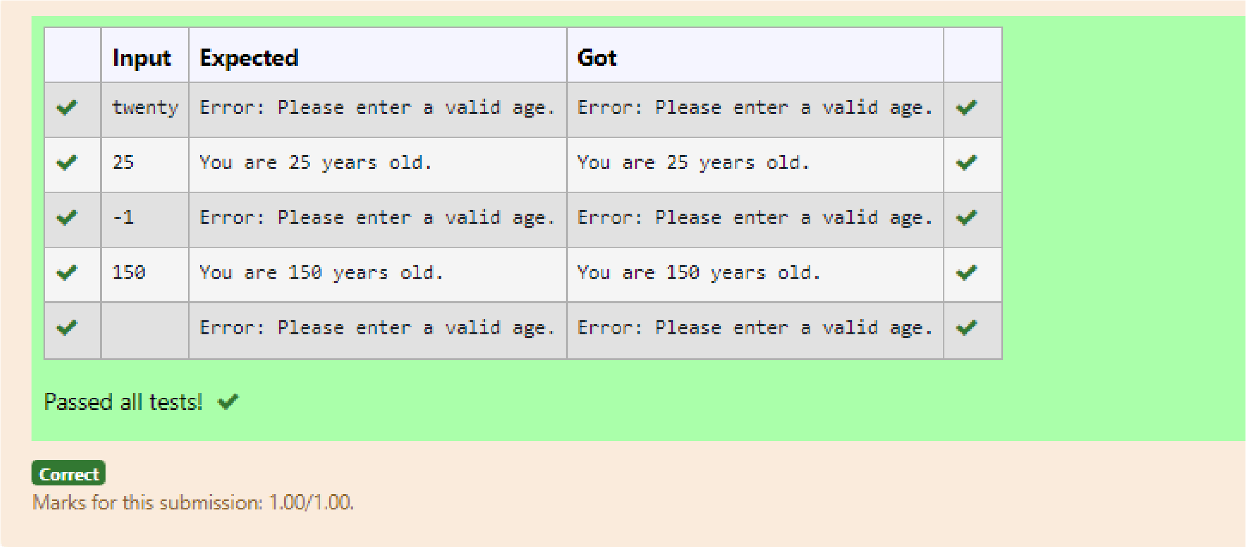
For example:

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

PROGRAM:-

```
def age(n):
    l=[str(i) for i in range(10)]
    flag=0
    for i in n:
        if i not in l:
            flag=1
            break
    if flag:
         return "Error: Please enter a valid age."
    else:
        return f"You are {n} years old."
while True :
    try :
        n=input()
        print(age(n))
        break
    except EOFError:
        print("Error: Please enter a valid age.")
        break
```

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. : 11.2 **Date: 6-6-24**

Register No.: 231501140 **Name: Sai Senthil .M**

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.

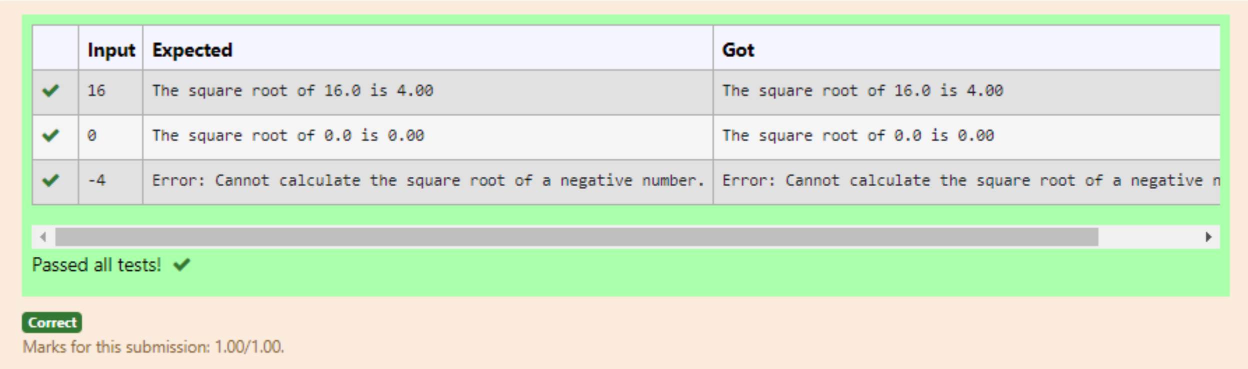
For example:

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

PROGRAM:-

```
import math


def safe_square_root():
    try:
        num = input()
        num = float(num)
        if num < 0:
            raise ValueError("Cannot calculate the square root of a negative number.")
        result = math.sqrt(num)
        print(f"The square root of {num} is {result:.2f}")
    except ValueError as e:
        if str(e) == "could not convert string to float: '{}'".format(num):
            print("Error: could not convert string to float")
        else:
            print(f"Error: {e}")
safe_square_root()
```

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 n

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Ex. No. : 11.3 Date: 6-6-24

Register No.: 231501140 Name: Sai Senthil .M

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.

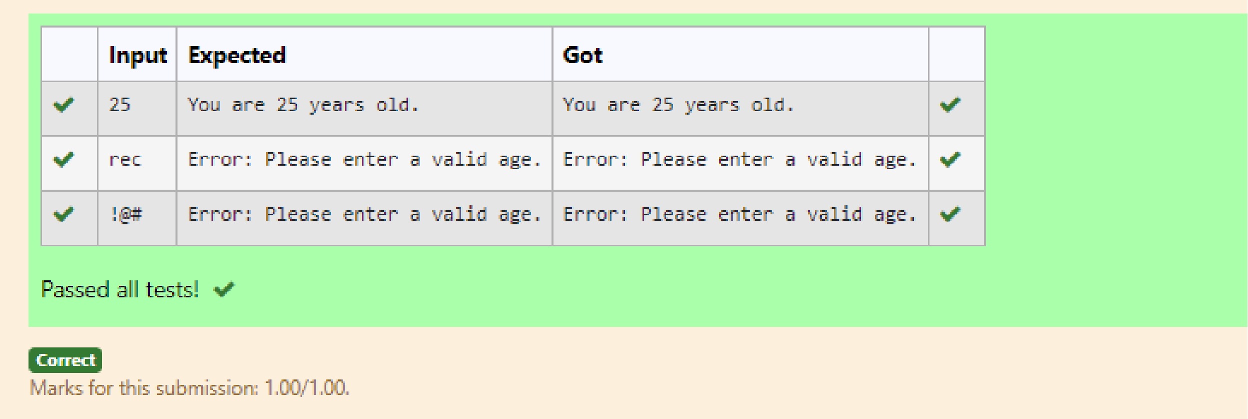
For example:

Input	Result
`25`	`You are 25 years old.`
`rec`	`Error: Please enter a valid age.`
`-5`	`Error: Please enter a valid age.`

PROGRAM:-

```
def age(n):
    l=[str(i) for i in range(10)]
    flag=0
    for i in n:
        if i not in l:
            flag=1
            break
    if flag:
         return "Error: Please enter a valid age."
    else:
        return f"You are {n} years old."
while True :
    try :
        n=input()
        print(age(n))
        break
    except EOFError:
        print("Error: Please enter a valid age.")
        break
```

OUTPUT:-



	Input	Expected	Got	
✓	25	`You are 25 years old.`	`You are 25 years old.`	✓
✓	rec	`Error: Please enter a valid age.`	`Error: Please enter a valid age.`	✓
✓	!@#	`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: 6-6-24**

Register No.: 231501140 **Name: Sai Senthil .M**

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.

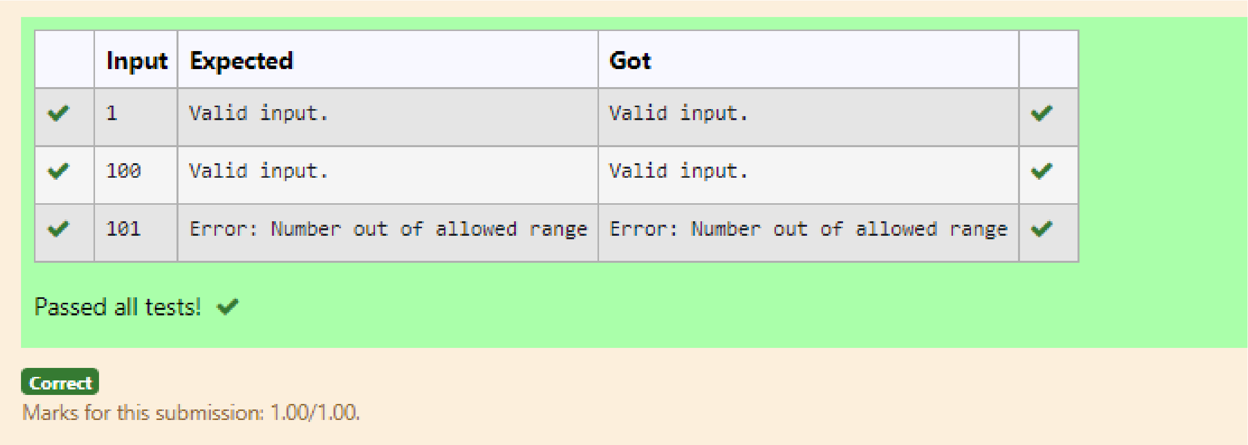
For example:

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

PROGRAM:-

```
def validate_input():
    try:
        user = int(input())
        if 1 <= user <= 100:
            print("Valid input.")
        else:
            print("Error: Number out of allowed range")
    except ValueError:
        print("Error: invalid literal for int()")
validate_input()
```

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! ✓

Correct

Marks for this submission: 1.00/1.00.

Ex. No. : 11.5 **Date: 6-6-24**

Register No.: 231501140 **Name: Sai Senthil .M**

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`

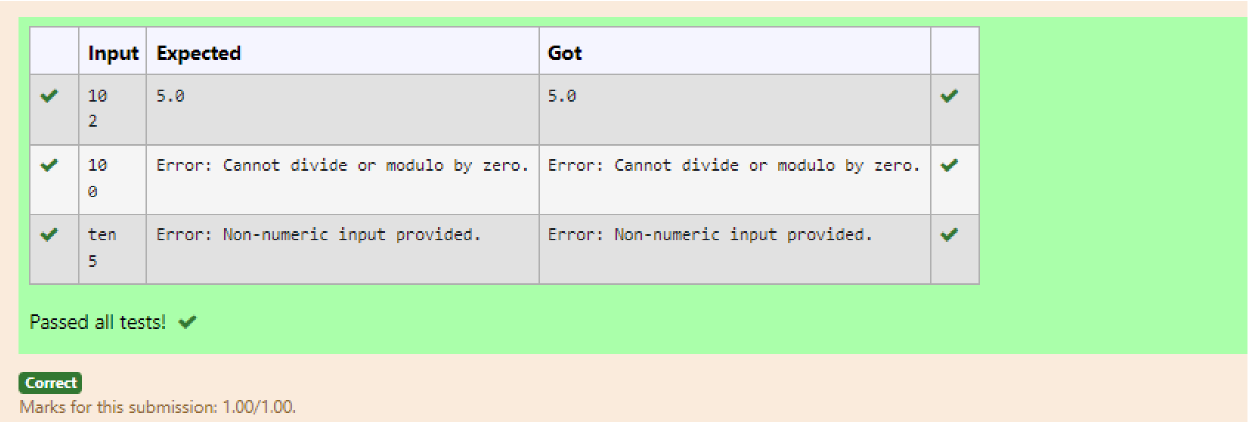
`10`
`0`
	`Error: Cannot divide or modulo by zero.`

`ten`
`5`
	`Error: Non-numeric input provided.`

PROGRAM:-

```
while True:

    try:

        num1 = float(input(""))

        num2 = float(input(""))

        result = num1 / num2

        print(f"{result}")

        break

    except ValueError:

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

        break

    except ZeroDivisionError:

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

        break
```

OUTPUT:-



	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.	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.