**11 - Exceptions**

**Ex. No. : 11.1 Date:** **1/6/24**

**Register No.: 231501015 Name: Aniruth S V**

**Safe Square Root**

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\_sqrt():

try:

num = float(input())

if num < 0:

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

else:

result = math.sqrt(num)

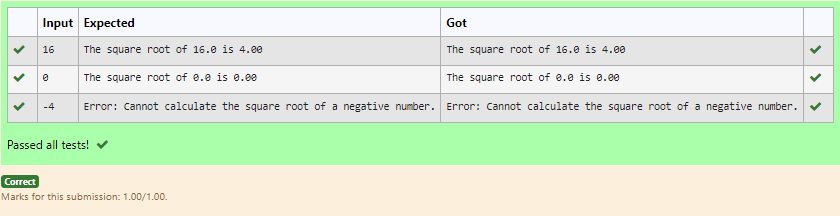
print("The square root of", num, "is", "{:.2f}".format(result))

except ValueError:

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

safe\_sqrt()

**Output:**

****

**Ex. No. : 11.2 Date:** **1/6/24**

**Register No.: 231501015 Name: Aniruth S V**

**Valid Age**

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.

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

**For example:**

**Program:**

def age():

try:

ag=int(input())

if ag<0:

raise ValueError("Error: Please enter a valid age.")

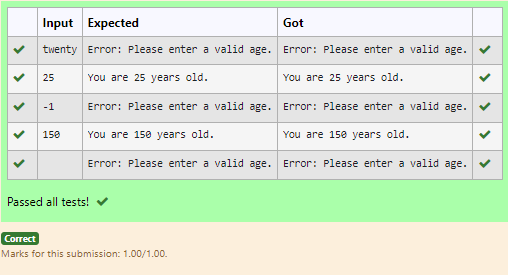
print("You are {} years old.".format(ag))

except Exception as e:

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

age()

**Output:**

****

**Ex. No. : 11.3 Date:** **1/6/24**

**Register No.: 231501015 Name: Aniruth S V**

**Valid Integer**

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:**

try:

n=int(input())

if n>=1:

print("You are",n,"years old.")

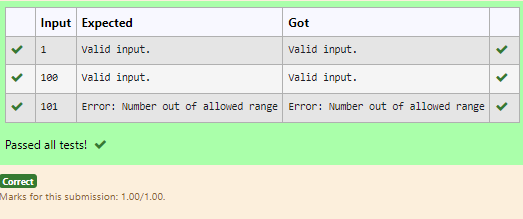
else:

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

except:

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

**Output:**

****

**Ex. No. : 11.4 Date:** **1/6/24**

**Register No.: 231501015 Name: Aniruth S V**

**Safe Division and Modulo**

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 the division 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. |

**Program:**

def operations():

try:

num1=int(input())

num2=int(input())

div=num1/num2

mod=num1%num2

print("Division result:", div)

print("Modulo result:", mod)

except ValueError:

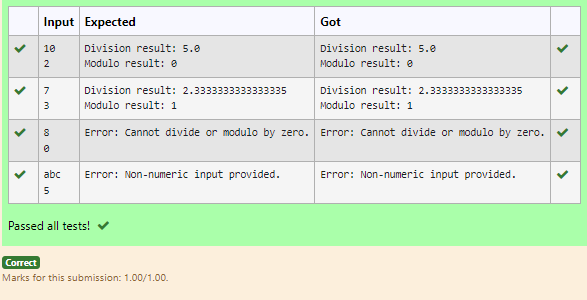
print("Error: Non-numeric input provided.")

except ZeroDivisionError:

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

operations()

**Output:**

****

**Ex. No. : 11.5 Date:** **1/6/24**

**Register No.: 231501015 Name: Aniruth S V**

**Safe Division**

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:**

def div():

try:

n1=input()

n2=input()

n1=float(n1)

n2=float(n2)

quo=n1/n2

print(quo)

except ValueError:

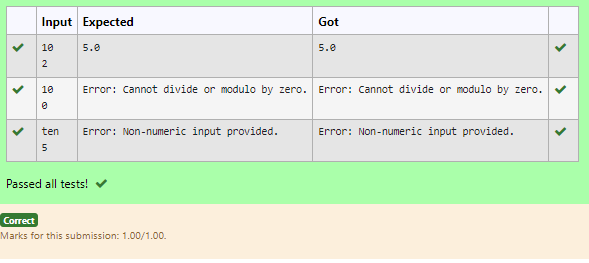
print("Error: Non-numeric input provided.")

except ZeroDivisionError:

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

div()

**Output:**

****