**Ans to the question number 1**

**Write a Python script that does the following**:

* + Prints the message: "Welcome to Python Programming!".
  + Declares three variables: an integer, a float, and a string. Assign values to these variables.
  + Displays the data type of each variable using the type () function.
  + Uses a single-line comment to describe each operation.

**Answer:**

# Printing the welcome message

print ("Welcome to Python Programming!")

# Declaring variables

integer\_var = 10 # Integer variable

float\_var = 20.5 # Float variable

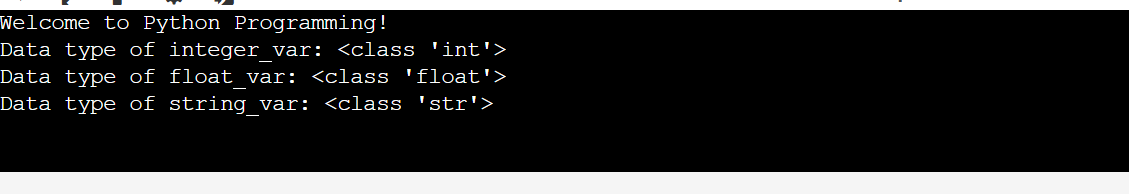
string\_var = "Python" # String variable

# Displaying the data types of each variable

print ("Data type of integer\_var:", type(integer\_var))

print ("Data type of float\_var:", type(float\_var))

print ("Data type of string\_var:", type(string\_var))

**Output-**

**Ans to the question number 2**

**Write a Python script to solve the following problem:**

* + Take two numbers as input from the user.
  + Perform and display the results of the following operations:
  + Addition, Subtraction, Multiplication, Division, and Modulus.
  + Check if the first number is greater than the second and display the result using a relational operator.
  + Use a logical operator to check if both numbers are positive and display the result.

**Answer:**

# Taking two numbers as input

num1 = float (input ("Enter the first number: "))

num2 = float (input ("Enter the second number: "))

# Performing arithmetic operations

print ("Addition:", num1 + num2) # Addition

print ("Subtraction:", num1 - num2) # Subtraction

print ("Multiplication:", num1 \* num2) # Multiplication

print ("Division:", num1 / num2 if num2! = 0 else "Undefined (division by zero)") # Division

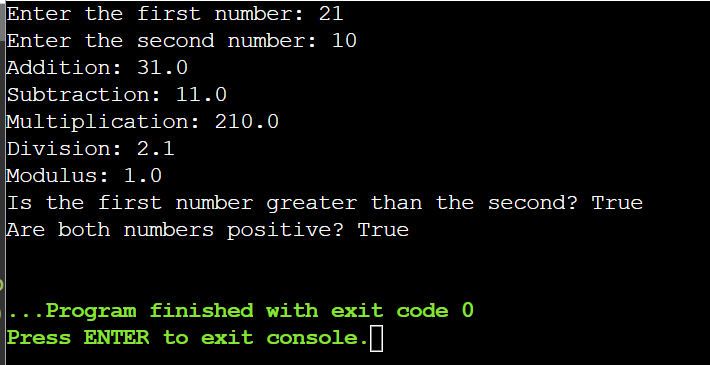
print ("Modulus:", num1 % num2 if num2! = 0 else "Undefined (division by zero)") # Modulus

# Relational operator

print ("Is the first number greater than the second?", num1 > num2)

# Logical operator

print ("Are both numbers positive?", num1 > 0 and num2 > 0)

**Output-**

**Ans to the question number 3**

**Write a Python program to evaluate a grade:**

* + Take a percentage as input from the user.
  + Determine the grade using the following criteria:
  + A: 90% and above
  + B: 80%-89%
  + C: 70%-79%
  + D: 60%-69%
  + F: Below 60%
  + Use if-elif-else statements to implement the logic.
  + Use indentation properly and ensure the program handles invalid input gracefully.

**Answer:**

# Taking percentage as input

try:

percentage = float (input ("Enter your percentage: "))

# Determining the grade

if percentage >= 90:

print ("Grade: A")

elif percentage >= 80:

print ("Grade: B")

elif percentage >= 70:

print ("Grade: C")

elif percentage >= 60:

print ("Grade: D")

elif percentage >= 0: # Ensures negative percentages are not accepted

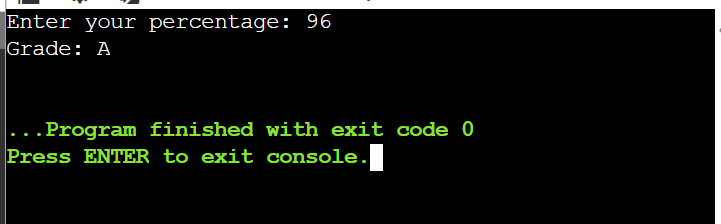
print ("Grade: F")

else:

print ("Invalid input. Please enter a positive percentage.")

except Value Error:

print ("Invalid input. Please enter a numeric value.")

**Output-**

**Ans to the question number 4**

**Write a Python script to print a multiplication table:**

* + Take an integer as input from the user.
  + Print the multiplication table for the given number up to 10.
  + Use a while loop to generate the table.
  + Add a feature to break the loop if the user enters a negative number.

**Answer:**

# Taking an integer as input

num = int (input ("Enter a number to generate its multiplication table (negative to quit): "))

# Using a while loop

if num < 0:

print ("Exiting the program.")

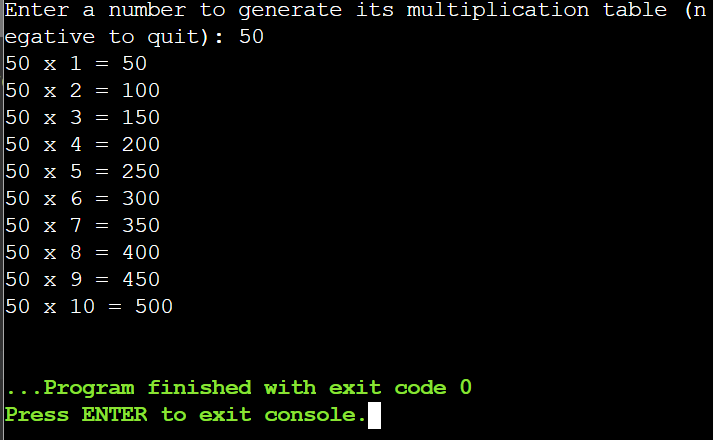
else:

i = 1

while i <= 10:

print(f"{num} x {i} = {num \* i}"

i += 1

**Output-**

**Ans to the question number 5**

**Write a Python script to display a pattern:**

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

* + Use nested loops to generate the pattern.
  + Include comments explaining the role of each loop.

**Answer:**

# Displaying a pattern using nested loops

rows = 5 # Number of rows in the pattern

for i in range (1, rows + 1): # Outer loop for rows

for j in range(i): # Inner loop for columns

print ("\*", end=" ") # Print stars in the same line

print () # New line after each row

**Output-**

