SECTION 1

Problem Statement : Write a program that takes your name and age as input and print a greeting like : “Hello John,you are 20 years old.”

Algorithm :

Step 1 : Start

Step 2 :enter their name and store it in the variable name.

Step 3 :enter their age and store it in the variable `age`.

Step 4 :Check if the `age` contains only digits using the `.isdigit()` method:

Step 5 : If True:Display the message:

`"Hello [name], you are [age] years old."`

Step 6 :If False:Display the message:

`"Please enter a valid age."`

Step 8 :End

Pseudo code :

```
START
  INPUT name
  INPUT age
  IF age contains only digits THEN
     DISPLAY "Hello [name], you are [age] years old."
  ELSE
     DISPLAY "Please enter a valid age."
END
```

Code :

```
name = input("Enter your name: ")
age = input("Enter your age: ")
if age.isdigit():
   print("Hello", name + ", you are", age, "years old.")
else:
   print("Please enter a valid age.")
```

Test cases	Input	Expected Out put
TC1	name:John age:20	Hello John, you are 20 years old

TC2	name:K age:-30	Enter a valid code
TC3	name:Raj age:ten	Enter a valid code

TC1:

```
Output
Enter your name: John
Enter your age: 20
Hello John, you are 20 years old.
```

TC2 :

```
Output
Enter your name: K
Enter your age: -30
Please enter a valid age.
```

TC3 :

```
Output
Enter your name: Raj
Enter your age: ten
Please enter a valid age.
```

Observation : The program works correctly when a valid name and numeric age are entered. It displays the greeting as expected. If the age is not a number (like "twenty" or "25.5"), it shows a message asking for valid input. The name can include letters, numbers, or symbols. If the name is left blank, the message still prints but may look incomplete. The program handles wrong inputs safely without crashing

Problem Statement 2: Take two numbers as input (string), convert them to integer and print their sum, difference and product .

Algorithm :

Step 1 : Start

Step 2 : Prompt the user to enter the first number and store it as `num1` (string).

Step 3 : Prompt the user to enter the second number and store it as `num2` (string).

Step 4 : Convert `num1` to an integer and store in variable `a`.

Step 5 : Convert `num2` to an integer and store in variable `b`.

Step 6 : Compute sum of a and b and store in `sum_result`.

Step 7 : Compute difference (`a - b`) and store in `diff_result`.

Step 8 : Compute product (`a * b`) and store in `prod_result`.

Step 9 : Display `sum_result`, `diff_result`, and `prod_result`.

Step
10 : End

Pseudo code :

```
START
DISPLAY "Enter the first number:"
READ num1
DISPLAY "Enter the second number:"
READ num2
SET a = convert num1 to integer
SET b = convert num2 to integer
SET sum_result = a + b
SET diff_result = a - b
SET prod_result = a * b
DISPLAY "Sum: ", sum_result
DISPLAY "Difference: ", diff_result
DISPLAY "Product: ", prod_result
END
```

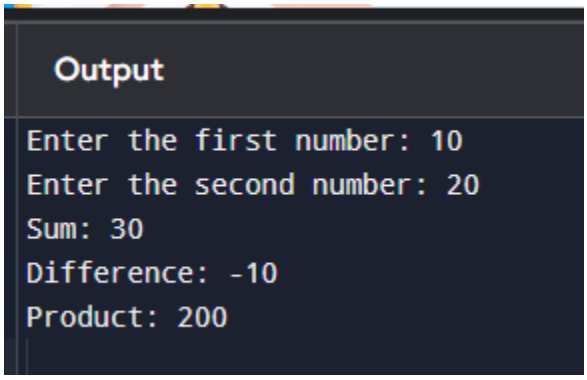
Code:

```
num1 = input("Enter the first number: ")
num2 = input("Enter the second number: ")

  a = int(num1)
  b = int(num2)
```

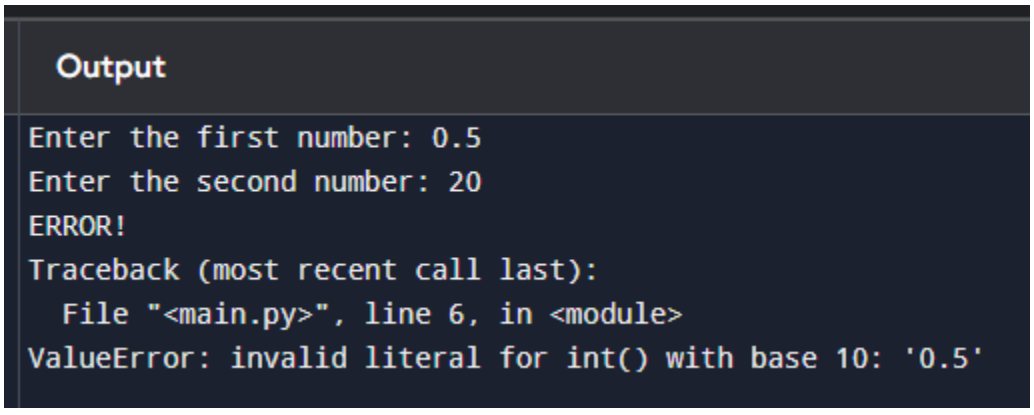
```
sum_result = a + b
diff_result = a - b
prod_result = a * b

print("Sum:", sum_result)
print("Difference:", diff_result)
print("Product:", prod_result)
```

Test cases	Input	Expected Out put
TC1	Num1:10 num2:20	Sum:30 Diff:-10 Prod:200
TC2	Num1:0.5 num2:20	Error
TC3	Num1:am num2:10	Error

TC1 :



TC2 :



TC3 :

```
Output
Enter the first number: am
Enter the second number: 10
ERROR!
Traceback (most recent call last):
  File "<main.py>", line 6, in <module>
ValueError: invalid literal for int() with base 10: 'am'
```

Observation :

The program takes two inputs from the user using the `input()` function. Since all inputs from `input()` are received as strings, the program checks whether the entered values are valid numbers using the If the input is not a number, the program displays an error message and stops running. Once both inputs are converted to integers, the program calculates and displays their sum, difference, and product. This ensures that the program handles only valid numeric inputs and avoids any runtime errors.

Problem Statement 3: Identify the data type of the following inputs in your language of choice : "123",123,123.45,True,"Hello"..

Algorithm :

Step 1 : Start.
Step 2 :Take input from the user as a single string, with values separated by commas.
Step 3 :Split the string by commas to get a list of values.
Step 4 :For each value in the list:
Step 5 :Check if the value is `"true"` or `"false"`
Step 6 : If yes, print type as `bool`.
Step 7 :Else, check if the value is a number using `.isdigit()`:
Step 8 :If it contains a dot (.), it's a float.
Step 9 :Otherwise, it's an int.
Step 10 : If none of the above, it's a str.
Step 11 :End

Pseudo code :

```
Start
Ask the user to enter values separated by commas
Store the input in a variable called user_input
Split the user_input by commas and store the parts in a list called values
For each value in values:
    Remove extra spaces from the value
    If value is "true" or "false"
        Print: value is of type bool
    Else if value has only digits:
        If value has a dot:
            Print: value is of type float
        Else:
            Print: value is of type int
    Else:
        Print: value is of type string
End
```

Code:

```
user_input = input("Enter values separated by commas: ")
values = user_input.split(",")
for value in values:
    if value.lower() == "true" or value.lower() == "false":
        print(value, "is of type bool")
    elif value.replace('.', ",).isdigit():
        if '.' in value:
            print(value, "is of type float")
        else:
            print(value, "is of type int")
    else:
        print(value, "is of type str")
```

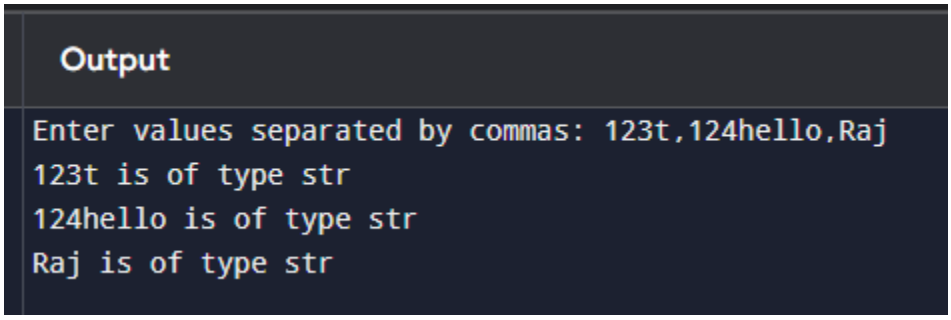
Test cases	Input	Expected Out put
TC1	Enter values separated by commas : “123”,123,123.45,True,”Hello”	123.45 is of type float 123 is of type int "123" is of type str true is of type bool "hello" is of type str
TC2	Enter values separated by commas :123t,1234hello,Raj	123t is of type str 124hello is of type str Raj is of type str
TC3	Enter values separated by commas :-124,"hello",23.5	-124 is of type str "hello" is of type str 23.5 is of type float

TC1 :

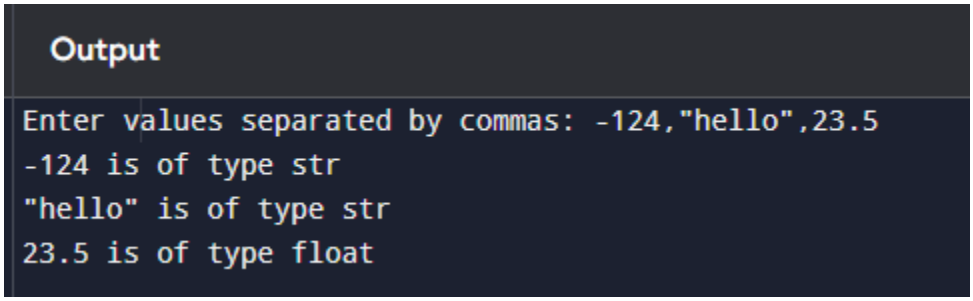
Output

```
Enter values separated by commas: 123.45,123,"123",true,"hello"
123.45 is of type float
123 is of type int
"123" is of type str
true is of type bool
"hello" is of type str
```

TC2 :



TC3 :



Observation :

The program is designed to identify the data types of multiple values entered by the user as a comma-separated string. It processes each value and checks if it is a boolean, float, integer, or string. Boolean values are correctly identified by comparing them to "true" or "false" in a case-insensitive manner. Float values are detected by checking for a decimal point in the number, and values that do not match these conditions are treated as strings.Overall, the program works well for simple inputs and provides a basic understanding of how to determine data types. It is beginner-friendly and useful for learning string handling and conditional statements in Python. However, to handle more complex cases accurately, the code can be improved using proper numeric conversions with error handling.

Problem Statement 4 : write the program that converts celsius to fahrenheit using a variable and formula F=(C*9/5) =32 for this give the python code.

Algorithm :

Step 1 : Start
Step 2 : Declare a variable to store Celsius temperature
Step 3 : Read the Celsius temperature from the user
Step 4 : Apply the formula: Fahrenheit = (Celsius × 9 / 5) + 32
Step 5 : Display the Fahrenheit temperature
Step 6: End

Pseudo code :

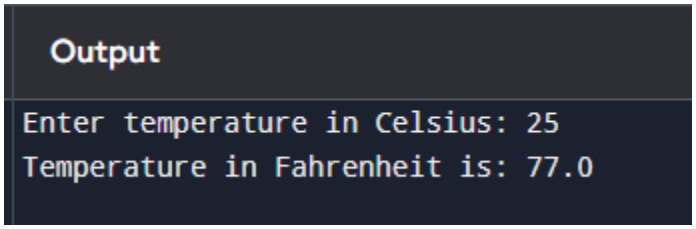
START
DISPLAY "Enter temperature in Celsius: "
READ Celsius
Fahrenheit ← (Celsius × 9 / 5) + 32
DISPLAY "Temperature in Fahrenheit is: ", Fahrenheit
END

Code:

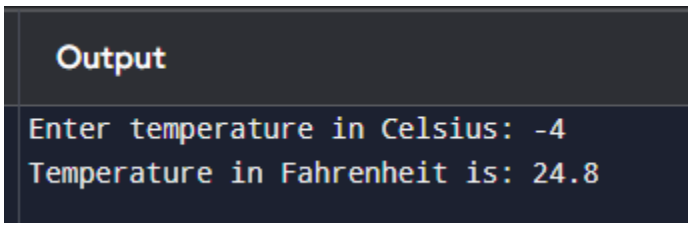
```
celsius = float(input("Enter temperature in Celsius: "))
fah = (celsius * 9 / 5) + 32
print("Temperature in Fah is:", fah)
```

Test cases	Input	Expected Out put
TC1	Enter the temperature : 25	Temperature in Fahrenheit is: 77.0
TC2	Enter the temperature : -4	Temperature in Fahrenheit is: 24.8
TC3	Enter the temperature : 4.5	Temperature in Fahrenheit is: 40.1

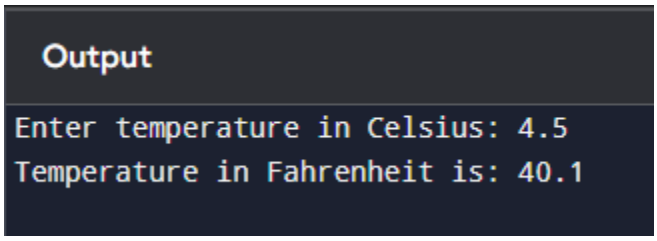
TC1 :



TC2 :



TC3 :



Observation :

This program takes a temperature value in Celsius as input from the user. It then applies the standard conversion formula F = (C × 9 / 5) + 32 to calculate the equivalent temperature in Fahrenheit. The result is stored in a variable and displayed to the user. This program demonstrates the use of variables, arithmetic operations, and basic input/output handling in Python. It is useful for learning how to perform simple mathematical conversions through programming.

Problem Statement 5 : Create a simple calculator that perform +,-,*,/between two user provided numbers for this statement give the code

Algorithm :

Step 1 : Start the program.
Step 2 : Ask the user to enter the first number.
Step 3 : Ask the user to enter an operator (+, -, *, /).
Step 4 : Ask the user to enter the second number.
Step 5 : Check the operator:
    If +, add the numbers.
    If -, subtract the second number from the first.
    If *, multiply the numbers.
    If /, two if second number is not zero.
Step 6 : Show the result.
Step 7 : If operator is not valid, show an error message.
Step 8 : End the program

Pseudo code :

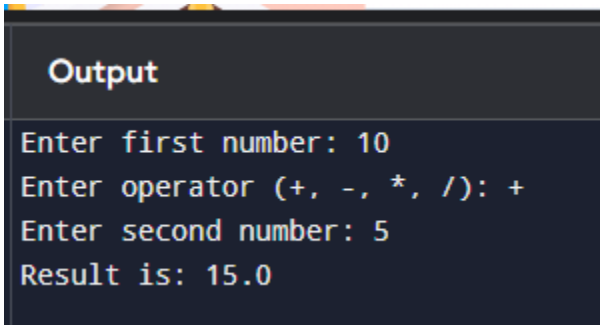
```
START
   READ number1
   READ operator (+, -, *, /)
   READ number2
   IF operator is "+"
      result ← number1 + number2
   ELSE IF operator is "-"
      result ← number1 - number2
   ELSE IF operator is "*"
      result ← number1 * number2
   ELSE IF operator is "/"
      IF number2 ≠ 0 THEN
         result ← number1 / number2
      ELSE
         PRINT "Cannot divide by zero"
   ELSE
      PRINT "Invalid operator"
   PRINT result
END
```

Code:

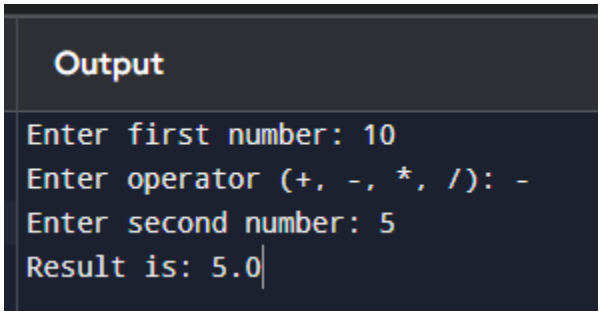
```
num1 = float(input("Enter first number: "))
op = input("Enter operator (+, -, *, /): ")
num2 = float(input("Enter second number: "))
if op == "+":
    print("Result:", num1 + num2)
elif op == "-":
    print("Result:", num1 - num2)
elif op == "*":
    print("Result:", num1 * num2)
elif op == "/":
    if num2 != 0:
        print("Result:", num1 / num2)
    else:
        print("Error: Cannot divide by zero.")
else:
    print("Invalid operator.")
```

Test cases	Input	Expected Out put
TC1	Enter first number: 10 Enter operator (+, -, *, /): + Enter second number: 5	Result is: 15.0
TC2	Enter first number: 10 Enter operator (+, -, *, /): - Enter second number: 5	Result is: 5
TC3	Enter first number: 10 Enter operator (+, -, *, /): / Enter second number: 0	Cannot divide by zero

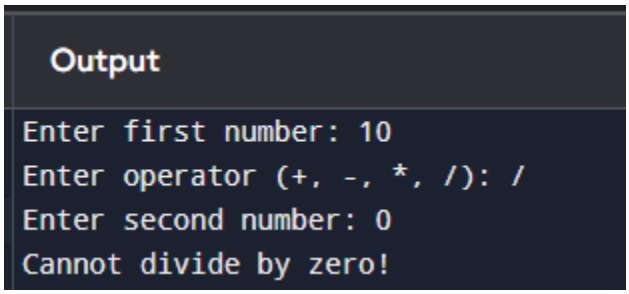
TC1 :



TC2 :



TC3 :



Observation :

The program successfully takes two numeric inputs and an operator from the user

It correctly performs the selected arithmetic operation: addition, subtraction, multiplication, or division.

It includes error handling for division by zero and invalid operators.

Problem Statement 6 : Accept a number from the user and print whether it is even or odd using if else for this statement give the code

Algorithm :

Step 1 : Start
Step 2 : Accept a number from the user.
Step 3 : Convert the input into an integer.
Step 4 : Check if the number is divisible by 2 (use modulus %).
Step 5 : If the remainder is 0, then print "Even".
Step 6 : Else, print "Odd".
Step 7 :End

Pseudo code :

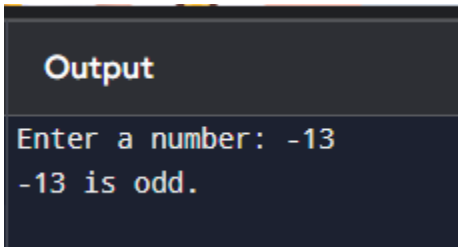
START
PROMPT user to enter a number
READ number
CONVERT number to integer and STORE in variable num
IF num MOD 2 equals 0 THEN
PRINT "num is even"
ELSE
PRINT "num is odd"
ENDIF
END

Code:

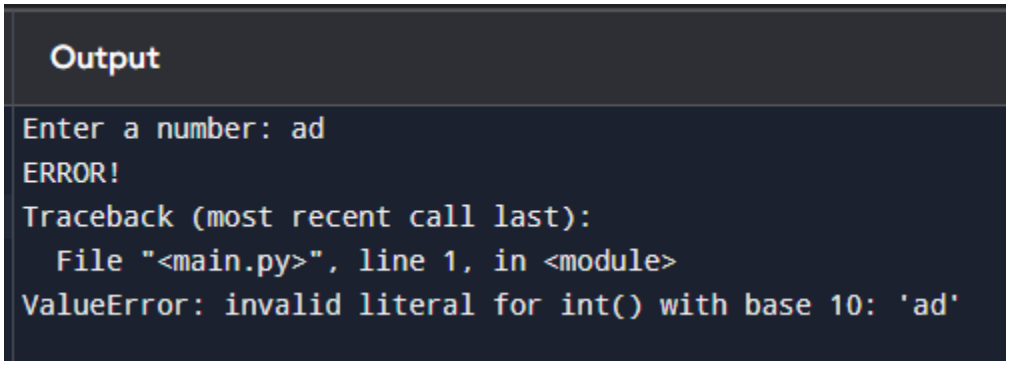
```
num = int(input("Enter a number: "))
if num % 2 == 0:
    print(num, "is even.")
else:
    print(num, "is odd.")
```

Test cases	Input	Expected Out put
TC1	Enter the number : -13	-13 is odd
TC2	Enter the number : ad	Error
TC3	Enter the number : 2.5	Error

TC1 :



TC2 :



TC3 :

```
Output
Enter a number: ERROR!
2.5
Traceback (most recent call last):
  File "<main.py>", line 1, in <module>
ValueError: invalid literal for int() with base 10: '2.5'
```

Observation :

The program accepts a number from the user and checks whether it is even or odd. It uses the modulus operator (%) to determine if the number is divisible by 2. If the remainder is zero, it prints that the number is even; otherwise, it prints that the number is odd. The program works correctly for both positive and negative integers. However, if a non-numeric value is entered, it will result in an error unless input validation is added.

Problem Statement 7: Accept a number from the user and print whether it is even or odd using if else for this statement give the code

Algorithm :

1. Start
2. Prompt the user to enter marks (0 to 100)
3. Read and convert the input into an integer
4. Check the value of marks using conditions:
    - If marks are between 90 and 100 → Grade A
    - If marks are between 80 and 89 → Grade B
    - If marks are between 70 and 79 → Grade C
    - If marks are between 60 and 69 → Grade D
    - If marks are between 0 and 59 → Grade F
    - Else → Invalid input
5. Display the appropriate grade
6. End

Pseudo code :

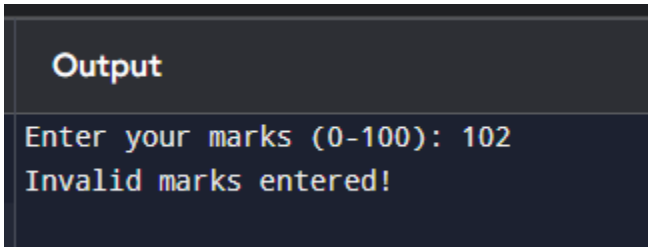
```
START
PROMPT user to enter marks
READ marks
CONVERT marks to integer
IF marks >= 90 AND marks <= 100 THEN
   PRINT "Grade: A"
ELSE IF marks >= 80 AND marks < 90 THEN
   PRINT "Grade: B"
ELSE IF marks >= 70 AND marks < 80 THEN
   PRINT "Grade: C"
ELSE IF marks >= 60 AND marks < 70 THEN
   PRINT "Grade: D"
ELSE IF marks >= 0 AND marks < 60 THEN
   PRINT "Grade: F"
ELSE
   PRINT "Invalid marks entered!"
ENDIF
END
```

Code:

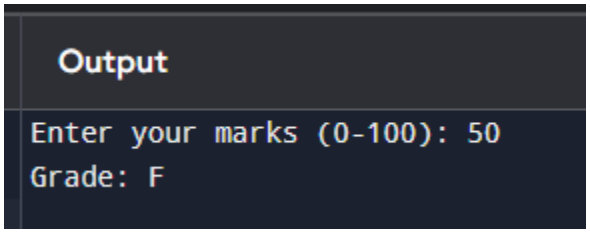
```
marks = int(input("Enter your marks (0-100): "))
if marks >= 90 and marks <= 100:
    print("Grade: A")
elif marks >= 80 and marks < 90:
    print("Grade: B")
elif marks >= 70 and marks < 80:
    print("Grade: C")
elif marks >= 60 and marks < 70:
    print("Grade: D")
elif marks >= 0 and marks < 60:
    print("Grade: F")
else:
    print("Invalid marks entered!")
```

Test cases	Input	Expected Out put
TC1	Enter the Marks : 102	Invalid Marks
TC2	Enter the Marks : 50	F
TC3	Enter the Marks : dh	Error

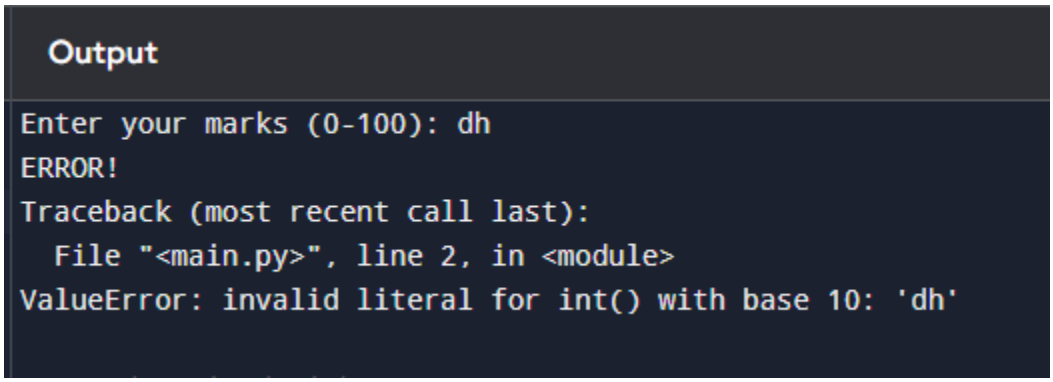
TC1 :



TC2 :



TC3 :



Observation :

The program accepts marks as input from the user and checks the range to assign the appropriate grade. It uses `if-elif-else` conditions to compare the marks with defined grade boundaries. Based on the value, it prints the grade: A for 90 and above, B for 80–89, C for 70–79, D for 60–69, and F for below 60. If the input is outside the range 0–100, it displays an "Invalid marks entered" message. The program works correctly for valid integer inputs within the specified range.

Problem Statement 8: Accept two numbers and print which is greater, or if they are equal.

Pseudo code :

```
Start

  Input num1

  Input num2

  If num1 > num2

        Print "First number is greater"

  Else if num2 > num1

        Print "Second number is greater"

  Else

        Print "Both numbers are equal"

End
```

Algorithm: steps

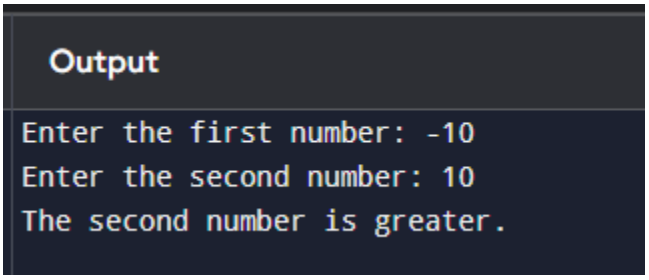
1. Start
2. Input two numbers: num1, num2
3. Compare num1 and num2:
    - If num1 > num2, print "First number is greater"
    - Else if num2 > num1, print "Second number is greater"
    - Else, print "Both numbers are equal"
4. End

Code :

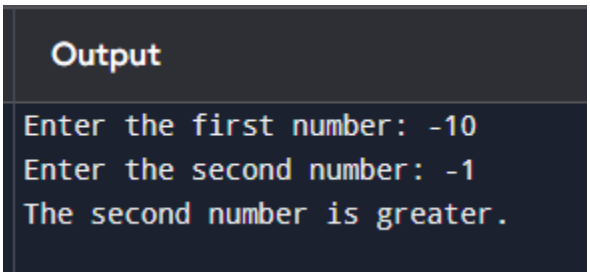
```
num1 = float(input("Enter the first number: "))
num2 = float(input("Enter the second number: "))
if num1 > num2:
   print("The first number is greater.")
elif num2 > num1:
   print("The second number is greater.")
else:
   print("Both numbers are equal.")
```

Test cases	Input	Expected Output
TC1	Enter the first number: -10 Enter the second number: 10	The second number is greater.
TC2	Enter the first number: -10 Enter the second number: -1	The second number is greater.
TC3	Enter the first number: 9 Enter the second number: 9	Both numbers are equal

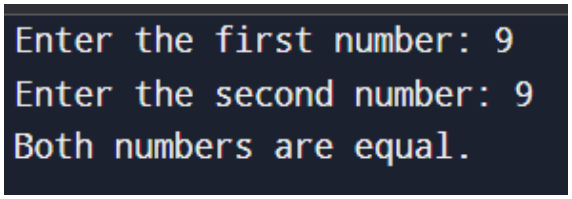
TC1:



TC2:



TC3:



Observation:

- The program takes two numbers as input from the user.
- It compares the numbers using conditional (if-elif-else) statements.
- It correctly identifies whether the first number is greater, the second is greater, or both are equal.
- The logic is simple and demonstrates basic number comparison effectively.

Problem Statement 9: Using a while loop, print numbers from 10 down to 1.

Pseudo code :

Start

Set num = 10

While num >= 1

Print num

Decrease num by 1

End

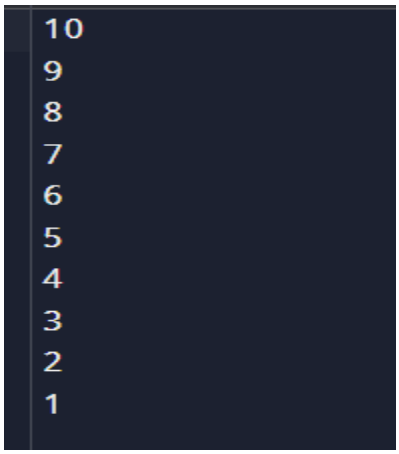
Algorithm: steps

1. Start
2. Initialize num to 10
3. While num is greater than or equal to 1, do:
    - Print num
    - Decrease num by 1
4. End

Code :

```
count =10
while count >= 1:
   print(count)
   count -= 1
```

Output:



Observation:

- The program uses a while loop to print numbers in reverse from 10 to 1.
- It initializes a counter and decreases it in each iteration.
- The loop stops when the number becomes less than 1.
- It demonstrates the use of decrementing control in a loop structure effectively.

Problem Statement 10 : Accept a number from the user and print its multiplication table up to 10 using a for loop.

Pseudo code :

Start

Input num

For i from 1 to 10

Print num × i = result

End

Algorithm: steps

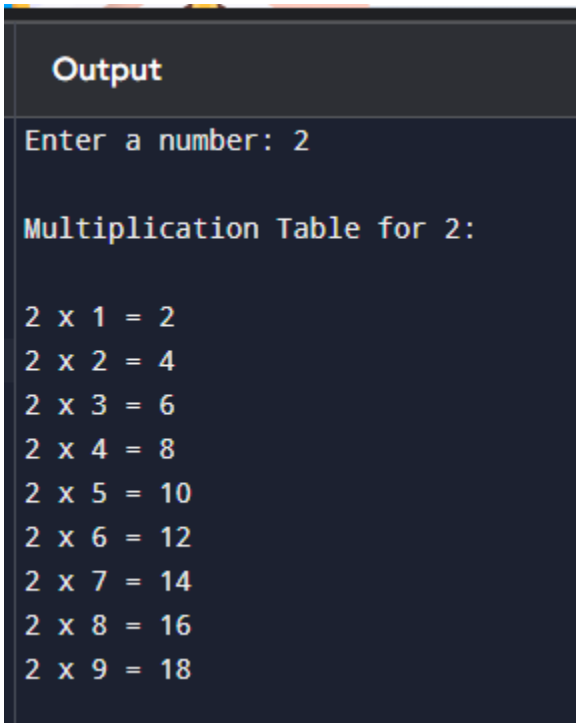
1. Start
2. Accept a number and store it in num
3. Loop i from 1 to 10
    - Calculate num × i
    - Print the result in the format "num x i = result"
4. End

Code :

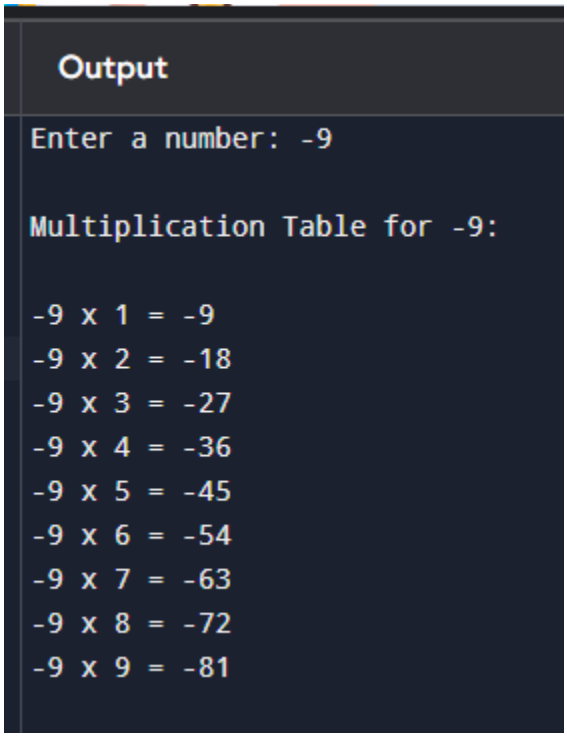
```
num = int(input("Enter a number: "))
print(f"\nMultiplication Table for {num}:\n")
for i in range(1, 10):
    print(f"{num} x {i} = {num * i}")
```

Test cases	Input	Expected Output
TC1	Enter the number : 2	Multiplication table for 2
TC2	Enter the number : -9	Multiplication table for -9
TC3	Enter the number : 3.0	Error

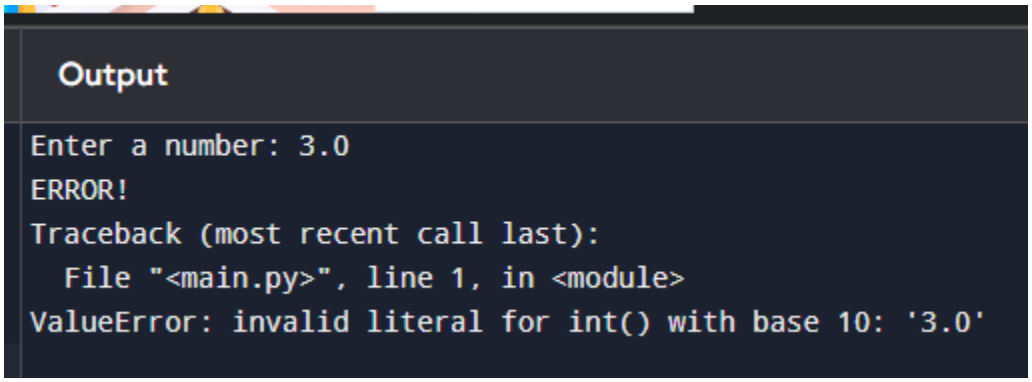
TC1 :



TC2:



TC3 :



Observation:

- The program generates a multiplication table for a given number using a for loop.
- It iterates from 1 to 10 and multiplies the input number by the loop counter.
- The output is displayed in a clear "num x i = result" format.
- It effectively demonstrates the use of loops and basic arithmetic operations.