Assignment - 1

1. Write a Python program to input three float numbers and find their sum and average.
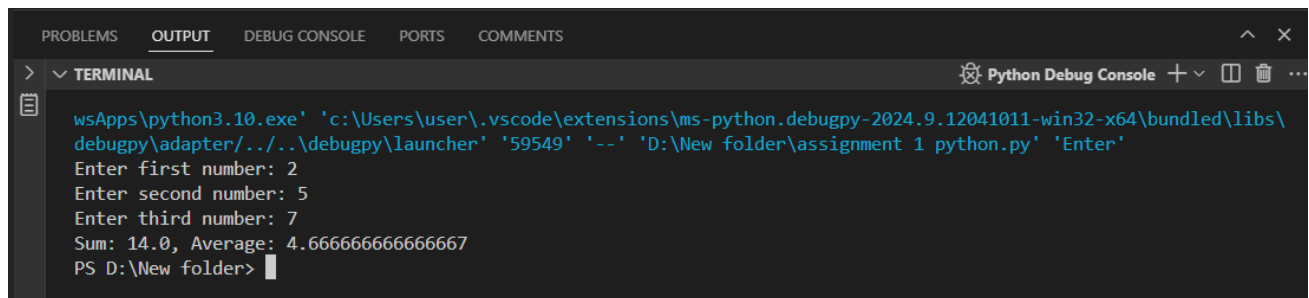
Code:

```python
num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))
num3 = float(input("Enter third number: "))

sum_of_numbers = num1 + num2 + num3

average = sum_of_numbers / 3

print(f"Sum: {sum_of_numbers}, Average: {average}")
```

Output:


```
PROBLEMS    OUTPUT    DEBUG CONSOLE    PORTS    COMMENTS                                                    ^  X
>  v TERMINAL                                                            Python Debug Console + v  ⬚ 🗑 ···

    wsApps\python3.10.exe' 'c:\Users\user\.vscode\extensions\ms-python.debugpy-2024.9.12041011-win32-x64\bundled\libs\
    debugpy\adapter/../..\debugpy\launcher' '59549' '--' 'D:\New folder\assignment 1 python.py' 'Enter'
    Enter first number: 2
    Enter second number: 5
    Enter third number: 7
    Sum: 14.0, Average: 4.666666666666667
    PS D:\New folder> ▮
```

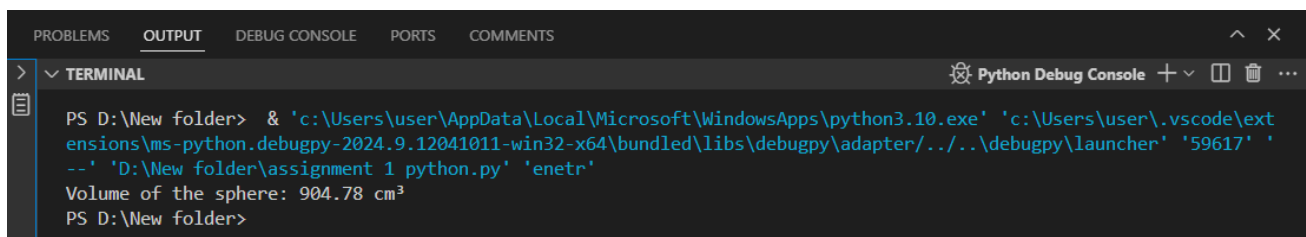2. **Write a program in Python to find the volume of a sphere with radius 6cm .**

Code:

```python
import math

radius = 6

volume = (4/3) * math.pi * radius**3

print(f"Volume of the sphere: {volume:.2f} cm³")
```

Output:

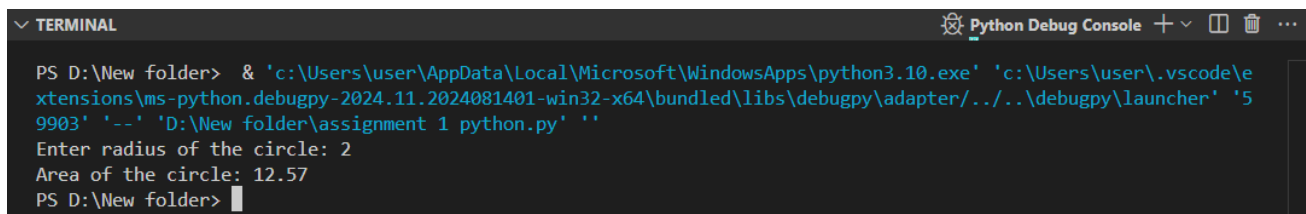## 3. Write a program in Python to find the area of a circle.

Code:

```python
import math

radius = float(input("Enter radius of the circle: "))

area = math.pi * radius**2

print(f"Area of the circle: {area:.2f}")
```

Output:

4. Write a program in Python to compute simple Interest.

Code:

```python
principal = float(input("Enter principal amount: "))
rate = float(input("Enter rate of interest: "))
time = float(input("Enter time in years: "))

simple_interest = (principal * rate * time) / 100

print(f"Simple Interest: {simple_interest}")
```

Output:

5. Write a program in Python to swap two numbers without using a third variable.

Code:

```python
a = int(input("Enter first number: "))
b = int(input("Enter second number: "))

a, b = b, a

print(f"After swapping: a = {a}, b = {b}")
```

Output:

6. Write a program in Python to convert any temperature from Celsius to Fahrenheit.

Code:

```python
celsius = float(input("Enter temperature in Celsius: "))

fahrenheit = (9/5) * celsius + 32

print(f"{celsius}°C is equal to {fahrenheit}°F")
```

Output:

7. Write a program in Python to convert a given number of days into days, month, year and week.

Code:
```python
days = int(input("Enter number of days: "))

years = days // 365
weeks = (days % 365) // 7
remaining_days = days % 7

print(f"{days} days is equal to {years} year(s), {weeks} week(s), and {remaining_days} day(s)")
```

Output:

8. Write a Python Program to find the gravitational force acting between two objects.

Code:
```python
G = 6.67430e-11
m1 = float(input("Enter mass of first object (in kg): "))
m2 = float(input("Enter mass of second object (in kg): "))
distance = float(input("Enter distance between the two objects (in meters): "))

force = G * (m1 * m2) / (distance ** 2)
```

```python
print(f"Gravitational force: {force} N")
```

Output:

```
v TERMINAL                                          Python Debug Console + v □ 🗑

 PS D:\New folder>  & 'c:\Users\user\AppData\Local\Microsoft\WindowsApps\python3.10.exe' 'c:\Users\user\.vscode\e
 xtensions\ms-python.debugpy-2024.11.2024081401-win32-x64\bundled\libs\debugpy\adapter/../..\debugpy\launcher' '6
 0124' '--' 'D:\New folder\assignment 1 python.py' ''
 Enter mass of first object (in kg): 5
 Enter mass of second object (in kg): 2
 Enter distance between the two objects (in meters): 7
 Gravitational force: 1.3621020408163264e-11 N
 PS D:\New folder> █
```

9. Write a program in Python to find if the given number is Even or Odd.

Code :

```python
num = int(input("Enter a number: "))

if num % 2 == 0:
    print(f"{num} is Even")
else:
    print(f"{num} is Odd")
```

Output:

```
v TERMINAL                                          Python Debug Console + v □ 🗑

 PS D:\New folder>  & 'c:\Users\user\AppData\Local\Microsoft\WindowsApps\python3.10.exe' 'c:\Users\user\.vscode\e
 xtensions\ms-python.debugpy-2024.11.2024081401-win32-x64\bundled\libs\debugpy\adapter/../..\debugpy\launcher' '6
 0146' '--' 'D:\New folder\assignment 1 python.py' ''
 Enter a number: 2
 2 is Even
 PS D:\New folder> █
```

10. Take input from the user; if it is greater than 15, print two times the difference; if it's less than 15, print four times the difference.

Code:

```python
num = int(input("Enter a number: "))

difference = abs(num - 15)

if num > 15:
    print(f"Two times the difference: {2 * difference}")
```

```
else:
    print(f"Four times the difference: {4 * difference}")
```

Output:

11. Write a program in Python to find the largest among three numbers.

Code:

```python
num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))
num3 = float(input("Enter third number: "))

largest = max(num1, num2, num3)

print(f"The largest number is: {largest}")
```
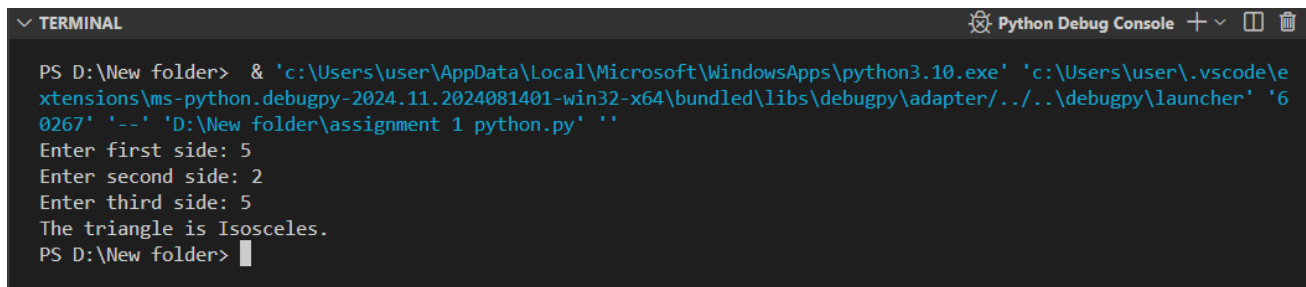
Output:

12. Write a program in Python to check if a triangle is equilateral, scalene, or isosceles.

Code:

```python
a = float(input("Enter first side: "))
b = float(input("Enter second side: "))
c = float(input("Enter third side: "))
```

```python
if a == b == c:
    print("The triangle is Equilateral.")
elif a != b and b != c and a != c:
    print("The triangle is Scalene.")
else:
    print("The triangle is Isosceles.")
```
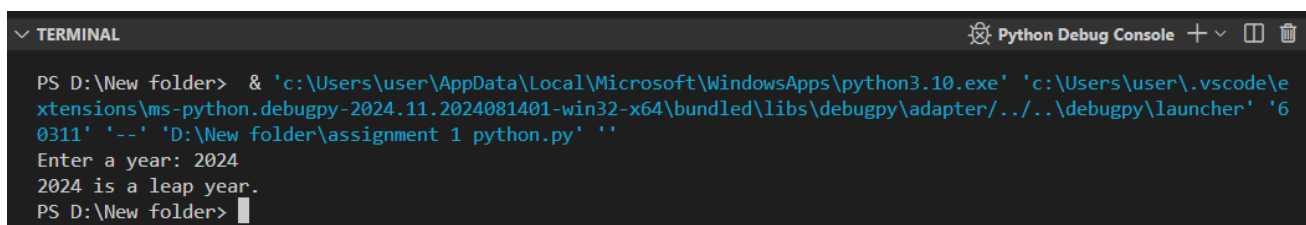
Output:

13. Write a program in Python to check if a given year is a leap year or not.

Code:

```python
year = int(input("Enter a year: "))

if (year % 4 == 0 and year % 100 != 0) or (year % 400 ==
0):
    print(f"{year} is a leap year.")
else:
    print(f"{year} is not a leap year.")
```

Output:

14. Take marks of a student in four different subjects, find average, and based on the average, determine the grade.
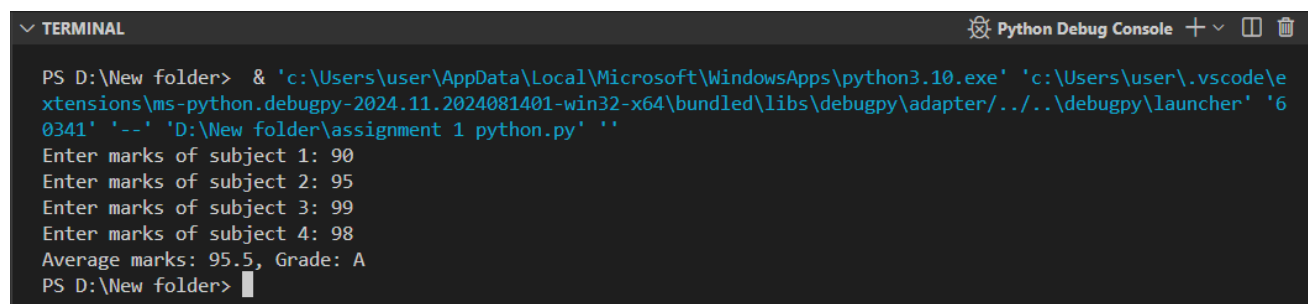
Code:

```python
marks1 = float(input("Enter marks of subject 1: "))
marks2 = float(input("Enter marks of subject 2: "))
marks3 = float(input("Enter marks of subject 3: "))
marks4 = float(input("Enter marks of subject 4: "))

average = (marks1 + marks2 + marks3 + marks4) / 4

if average >= 75:
    grade = 'A'
elif 60 <= average < 75:
    grade = 'B'
elif 40 <= average < 60:
    grade = 'C'
else:
    grade = 'D'

print(f"Average marks: {average}, Grade: {grade}")
```

Output:

```
∨ TERMINAL                                                          ⚙ Python Debug Console  + ∨  ⊡ 🗑

 PS D:\New folder>  & 'c:\Users\user\AppData\Local\Microsoft\WindowsApps\python3.10.exe' 'c:\Users\user\.vscode\e
 xtensions\ms-python.debugpy-2024.11.2024081401-win32-x64\bundled\libs\debugpy\adapter/../..\debugpy\launcher' '6
 0341' '--' 'D:\New folder\assignment 1 python.py' ''
 Enter marks of subject 1: 90
 Enter marks of subject 2: 95
 Enter marks of subject 3: 99
 Enter marks of subject 4: 98
 Average marks: 95.5, Grade: A
 PS D:\New folder>
```

15. Write a program in Python to implement a simple calculator.

Code:

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

operation = input("Enter operation (+, -, *, /): ")
```
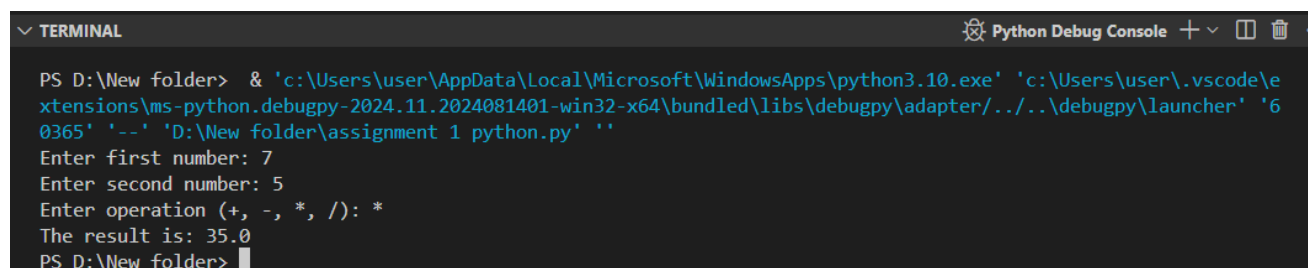
```python
if operation == '+':
    result = num1 + num2
elif operation == '-':
    result = num1 - num2
elif operation == '*':
    result = num1 * num2
elif operation == '/':
    if num2 != 0:
        result = num1 / num2
    else:
        result = "Division by zero is not allowed"
else:
    result = "Invalid operation"

print(f"The result is: {result}")
```

Output:


```
PS D:\New folder>  & 'c:\Users\user\AppData\Local\Microsoft\WindowsApps\python3.10.exe' 'c:\Users\user\.vscode\e
xtensions\ms-python.debugpy-2024.11.2024081401-win32-x64\bundled\libs\debugpy\adapter/../..\debugpy\launcher' '6
0365' '--' 'D:\New folder\assignment 1 python.py' ''
Enter first number: 7
Enter second number: 5
Enter operation (+, -, *, /): *
The result is: 35.0
PS D:\New folder>
```

16. Write a program in Python to find the roots of a Quadratic equation.

Code:

```python
import math
a = float(input("Enter coefficient a: "))
b = float(input("Enter coefficient b: "))
c = float(input("Enter coefficient c: "))
discriminant = b**2 - 4*a*c

if discriminant >= 0:
    root1 = (-b + math.sqrt(discriminant)) / (2 * a)
    root2 = (-b - math.sqrt(discriminant)) / (2 * a)
    print(f"The roots are real: {root1} and {root2}")
```
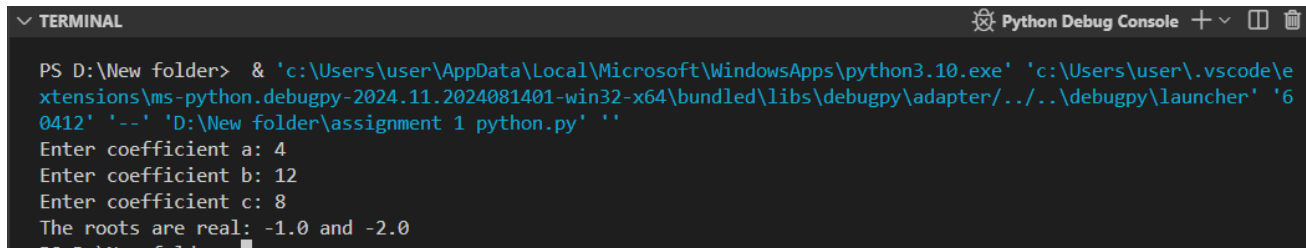
```python
else:
    real_part = -b / (2 * a)
    imaginary_part = math.sqrt(-discriminant) / (2 * a)
    print(f"The roots are complex: {real_part} ±
{imaginary_part}i")
```

Output:

```
∨ TERMINAL                                          ⚙ Python Debug Console + ∨ ⊓ 🗑

PS D:\New folder>  & 'c:\Users\user\AppData\Local\Microsoft\WindowsApps\python3.10.exe' 'c:\Users\user\.vscode\e
xtensions\ms-python.debugpy-2024.11.2024081401-win32-x64\bundled\libs\debugpy\adapter/../..\debugpy\launcher' '6
0412' '--' 'D:\New folder\assignment 1 python.py' ''
Enter coefficient a: 4
Enter coefficient b: 12
Enter coefficient c: 8
The roots are real: -1.0 and -2.0
DC D.\Naw faldar
```

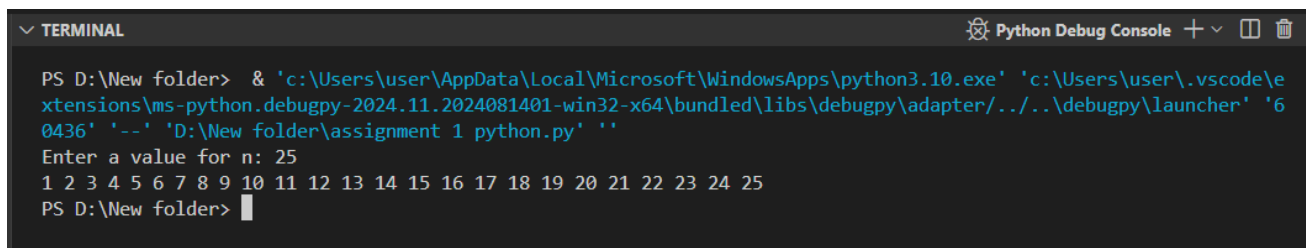## 17. Write a program in Python to print 1 to n.

Code:

```python
n = int(input("Enter a value for n: "))

for i in range(1, n + 1):
    print(i, end=" ")
```

Output:

```
∨ TERMINAL                                          ⚙ Python Debug Console + ∨ ⊓ 🗑

PS D:\New folder>  & 'c:\Users\user\AppData\Local\Microsoft\WindowsApps\python3.10.exe' 'c:\Users\user\.vscode\e
xtensions\ms-python.debugpy-2024.11.2024081401-win32-x64\bundled\libs\debugpy\adapter/../..\debugpy\launcher' '6
0436' '--' 'D:\New folder\assignment 1 python.py' ''
Enter a value for n: 25
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
PS D:\New folder>
```

18. Write a program in Python to print the sum of natural numbers.

Code:

```python
n = int(input("Enter a value for n: "))

sum_of_n = sum(range(1, n + 1))
```

```
print(f"Sum of first {n} natural numbers is: {sum_of_n}")
```

Output:

```
∨ TERMINAL                                                          ⚙ Python Debug Console  + ∨  ⊡  🗑

  PS D:\New folder>  & 'c:\Users\user\AppData\Local\Microsoft\WindowsApps\python3.10.exe' 'c:\Users\user\.vscode\e
  xtensions\ms-python.debugpy-2024.11.2024081401-win32-x64\bundled\libs\debugpy\adapter/../..\debugpy\launcher' '6
  0493' '--' 'D:\New folder\assignment 1 python.py' ''
  Enter a value for n: 52
  Sum of first 52 natural numbers is: 1378
  PS D:\New folder> ▮
```

19. Write a program in Python to print Sum of Digits.

Code:
```
num = int(input("Enter a number: "))
sum_of_digits = sum(int(digit) for digit in str(num))

print(f"Sum of digits: {sum_of_digits}")
```

Output:

```
∨ TERMINAL                                                          ⚙ Python Debug Console  + ∨  ⊡  🗑

  PS D:\New folder>  & 'c:\Users\user\AppData\Local\Microsoft\WindowsApps\python3.10.exe' 'c:\Users\user\.vscode\e
  xtensions\ms-python.debugpy-2024.11.2024081401-win32-x64\bundled\libs\debugpy\adapter/../..\debugpy\launcher' '6
  0525' '--' 'D:\New folder\assignment 1 python.py' ''
  Enter a number: 25
  Sum of digits: 7
  PS D:\New folder> ▮
```

20. Write a program in Python to print Factors of a Number.

Code:
```
num = int(input("Enter a number: "))

print(f"Factors of {num} are: ", end="")
for i in range(1, num + 1):
    if num % i == 0:
        print(i, end=" ")
```

Output:

21. Write a program in Python to print Reverse of a Number.

Code:

```python
num = int(input("Enter a number: "))

reversed_num = int(str(num)[::-1])

print(f"Reversed number: {reversed_num}")
```

Output:

22. Write a Python program to find Factorial of a given number.

Code:

```python
num = int(input("Enter a number: "))

factorial = 1
for i in range(1, num + 1):
    factorial *= i

print(f"Factorial of {num} is: {factorial}")
```

Output:

23. Write a Python program to check if a number is palindrome or not.

Code:

```python
num = int(input("Enter a number: "))

if str(num) == str(num)[::-1]:
    print(f"{num} is a palindrome.")
else:
    print(f"{num} is not a palindrome.")
```

Output:

24. Write a Python program to check if a number is prime or not.

Code:

```python
num = int(input("Enter a number: "))

if num > 1:
    for i in range(2, int(num**0.5) + 1):
        if num % i == 0:
            print(f"{num} is not a prime number.")
            break
    else:
        print(f"{num} is a prime number.")
else:
    print(f"{num} is not a prime number.")
```

Output:

TERMINAL   ⚙ Python Debug Console + ∨ ▯ 🗑

PS D:\New folder> & 'c:\Users\user\AppData\Local\Microsoft\WindowsApps\python3.10.exe' 'c:\Users\user\.vscode\e
xtensions\ms-python.debugpy-2024.11.2024081401-win32-x64\bundled\libs\debugpy\adapter/../..\debugpy\launcher' '6
0721' '--' 'D:\New folder\assignment 1 python.py' ''
Enter a number: 5
5 is a prime number.
PS D:\New folder> ▋

25. Write a Python program to check if a number is Armstrong or not.

Code:

```python
num = int(input("Enter a number: "))

num_of_digits = len(str(num))

armstrong_sum = sum(int(digit)**num_of_digits for digit in
str(num))

if num == armstrong_sum:
    print(f"{num} is an Armstrong number.")
else:
    print(f"{num} is not an Armstrong number.")
```

Output:

TERMINAL   ⚙ Python Debug Console + ∨ ▯ 🗑

PS D:\New folder> & 'c:\Users\user\AppData\Local\Microsoft\WindowsApps\python3.10.exe' 'c:\Users\user\.vscode\e
xtensions\ms-python.debugpy-2024.11.2024081401-win32-x64\bundled\libs\debugpy\adapter/../..\debugpy\launcher' '6
0745' '--' 'D:\New folder\assignment 1 python.py' ''
Enter a number: 153
153 is an Armstrong number.
PS D:\New folder> ▋

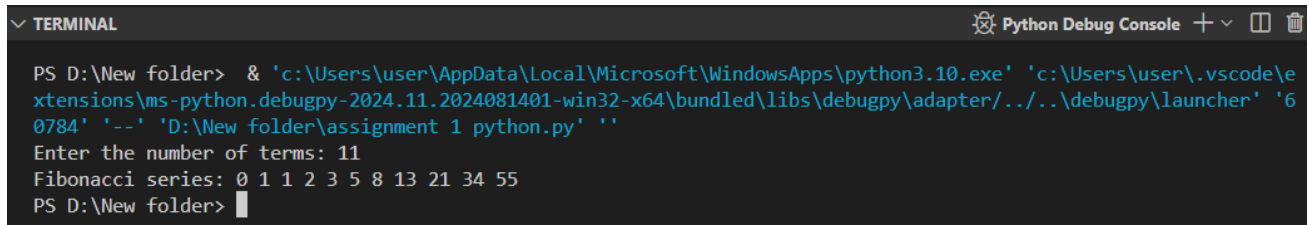## 26. Write a Python program to print Fibonacci series of n terms.

Code:

```python
n = int(input("Enter the number of terms: "))

a, b = 0, 1

print("Fibonacci series:", end=" ")
```

```python
for _ in range(n):
    print(a, end=" ")
    a, b = b, a + b
```

Output:

```
∨ TERMINAL                                                    Python Debug Console  +∨  ⬚  🗑

PS D:\New folder>  & 'c:\Users\user\AppData\Local\Microsoft\WindowsApps\python3.10.exe' 'c:\Users\user\.vscode\e
xtensions\ms-python.debugpy-2024.11.2024081401-win32-x64\bundled\libs\debugpy\adapter/../..\debugpy\launcher' '6
0784' '--' 'D:\New folder\assignment 1 python.py' ''
Enter the number of terms: 11
Fibonacci series: 0 1 1 2 3 5 8 13 21 34 55
PS D:\New folder> ▌
```
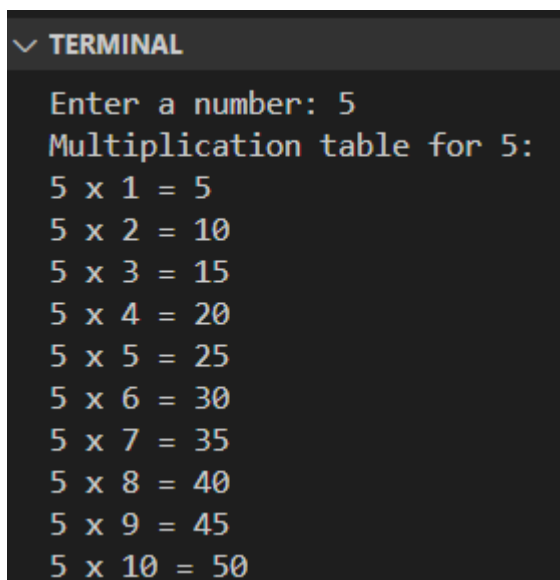
27. Write a Python program to implement multiplication table.

Code:

```python
num = int(input("Enter a number: "))

print(f"Multiplication table for {num}:")
for i in range(1, 11):
    print(f"{num} x {i} = {num * i}")
```

Output:

```
∨ TERMINAL

Enter a number: 5
Multiplication table for 5:
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
```

28. Write a Python program to take input from user in a list and print it.

Code:

```python
n = int(input("Enter number of elements in the list: "))

user_list = [input(f"Enter element {i+1}: ") for i in
range(n)]

print("The list is:", user_list)
```

Output:

```
∨ TERMINAL

  Enter element 1: 5
  Enter element 2: 2
  Enter element 3: 4
  Enter element 4: 6
  Enter element 5: 9
  Enter element 6: 7
  Enter element 7: 1
  The list is: ['5', '2', '4', '6', '9', '7', '1']
  PS D:\New folder> ▌
```

29. Write a Python program to find the average of n numbers using a list.

Code:
```python
n = int(input("Enter number of elements: "))

numbers = [float(input(f"Enter number {i+1}: ")) for i in
range(n)]

average = sum(numbers) / n

print(f"Average of the numbers is: {average}")
```

Output:

30. Write a Python program to find the largest element in the list.

Code:
```
n = int(input("Enter number of elements: "))
numbers = [float(input(f"Enter number {i+1}: ")) for i in
range(n)]
print(f"Largest element in the list is: {max(numbers)}")
```

Output:

31. Write a Python program to perform Linear search.

Code:
```
n = int(input("Enter number of elements: "))

numbers = [int(input(f"Enter element {i+1}: ")) for i in
range(n)]

search_element = int(input("Enter the element to search:
"))

found = False
for i in range(n):
    if numbers[i] == search_element:
```
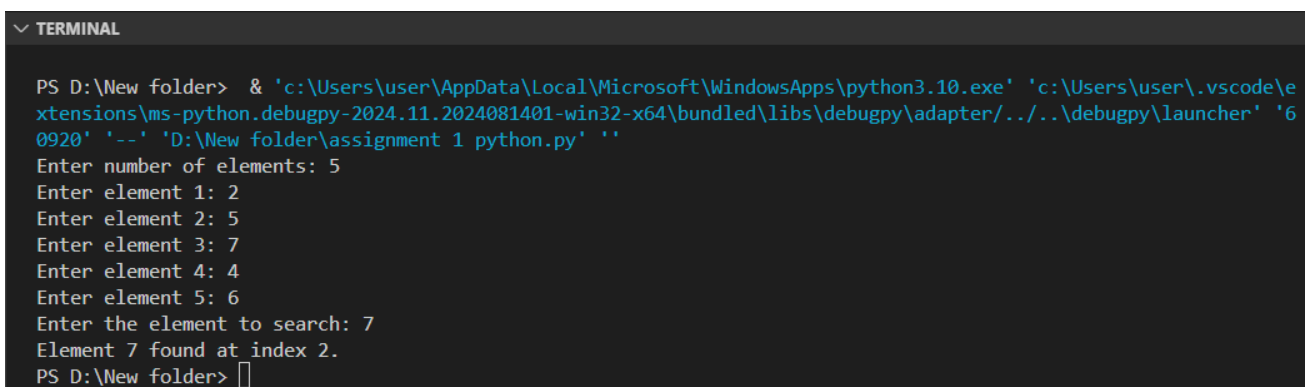
```
        found = True
        print(f"Element {search_element} found at index
{i}.")
        break


if not found:
    print(f"Element {search_element} not found in the
list.")
```

Output:

32. Write a program that accepts sequence of lines as input and prints the lines after making all characters in the sentence capitalized.

Code:
```
lines = []
while True:
    line = input("Enter a line (or type 'STOP' to end): ")
    if line == "STOP":
        break
    lines.append(line)


for line in lines:
    print(line.upper())
```

Output:

33. Write a program which accepts a sequence of comma-separated numbers from the console and generates a list and a tuple which contains every number.

Code:

```
input_string = input("Enter comma-separated numbers: ")

numbers_list = input_string.split(",")
numbers_tuple = tuple(numbers_list)

print("List:", numbers_list)
print("Tuple:", numbers_tuple)
```

Output:

34. Write a program that computes the net amount of a bank account based on a transaction log from console input.

Code:

```
net_amount = 0

while True:
    transaction = input("Enter transaction (or type 'STOP'
to end): ")
    if transaction == "STOP":
```

```
        break

    transaction_type, amount = transaction.split()
    amount = int(amount)

    if transaction_type == 'D':
        net_amount += amount
    elif transaction_type == 'W':
        net_amount -= amount

print(f"Net amount in the account: {net_amount}")
```

Output:

TERMINAL   Python Debug Console + ∨

 PS D:\New folder> & 'c:\Users\user\AppData\Local\Microsoft\WindowsApps\python3.10.exe' 'c:\Users\user\.vs
 code\extensions\ms-python.debugpy-2024.11.2024082901-win32-x64\bundled\libs\debugpy\adapter/../..\debugpy\
 launcher' '61436' '--' 'D:\New folder\assignment 1 python.py' ''
 Enter transaction (or type 'STOP' to end): D 300
 Enter transaction (or type 'STOP' to end): D 300
 Enter transaction (or type 'STOP' to end): W 200
 Enter transaction (or type 'STOP' to end): D 100
 Enter transaction (or type 'STOP' to end): STOP
 Net amount in the account: 500
 PS D:\New folder>

—---------------------------- x x x x ---------------------------------