Started on	Sunday, 30 March 2025, 8:17 PM
State	Finished
Completed on	Sunday, 30 March 2025, 8:32 PM
Time taken	15 mins 10 secs
Grade	**10.00** out of 10.00 (**100**%)

Question **1**

Correct

Mark 2.00 out of 2.00

Write a Python Program Using a recursive function to calculate the sum of a sequence
For example:

Input	Result
20	210
36	666
45	1035

Answer: (penalty regime: 0 %)

```
def sum(n):
    if n<=1:
        return n
    else:
        return n + sum(n-1)

num=int(input())

print(sum(num));
```

	Input	Expected	Got	
✔	20	210	210	✔
✔	36	666	666	✔
✔	45	1035	1035	✔
✔	58	1711	1711	✔
✔	65	2145	2145	✔

Passed all tests! ✔

Correct

Marks for this submission: 2.00/2.00.

Question **2**

Correct

Mark 2.00 out of 2.00

Write a python Program Using Recursive Function which calculates the value of a number multiplied by itself a certain number of times.

For example:

Input	Result
4 7	4 to the power of 7 is 16384
9 6	9 to the power of 6 is 531441

Answer: (penalty regime: 0 %)

```
# Recursive function to calculate power of a number
def power(base, exponent):
    # Base case: If exponent is 0, return 1 (as any number raised to the power of 0 i
    if exponent == 0:
        return 1
    # Recursive case: Multiply base by the result of the recursive call with exponent
    return base * power(base, exponent - 1)

# Input from the user
base = int(input())
exponent = int(input())

# Calculate the result using the recursive function
result = power(base, exponent)

# Display the result
print(f"{base} to the power of {exponent} is {result}")
```

	Input	Expected	Got	
✔	4 7	4 to the power of 7 is 16384	4 to the power of 7 is 16384	✔
✔	9 6	9 to the power of 6 is 531441	9 to the power of 6 is 531441	✔
✔	5 4	5 to the power of 4 is 625	5 to the power of 4 is 625	✔
✔	4 6	4 to the power of 6 is 4096	4 to the power of 6 is 4096	✔

Passed all tests! ✔

Correct

Marks for this submission: 2.00/2.00.

Question **3**

Correct

Mark 2.00 out of 2.00

Use recursion to write a Python function for determining if a string has more vowels than consonants return True otherwise False.

For example:

Input	Result
string	False

Answer: (penalty regime: 0 %)

```
    return vowel_count > consonant_count

# Current character
char = string[index].lower()

# Check if the character is a vowel or consonant
if char in 'aeiou':
    vowel_count += 1
elif char.isalpha():  # Check if the character is a letter and not a vowel
    consonant_count += 1

# Recursively call for the next character
return has_more_vowels(string, index + 1, vowel_count, consonant_count)

# Test the function with an example input
input_string = input();
result = has_more_vowels(input_string)

# Display the result
print(result)

```

	Input	Expected	Got	
✔	string	False	False	✔
✔	Saveetha	False	False	✔
✔	aeiousd	True	True	✔
✔	engineering	False	False	✔

Passed all tests! ✔

Correct

Marks for this submission: 2.00/2.00.

Question **4**

Correct

Mark 2.00 out of 2.00

Write a Python program to calculate the harmonic sum of n-1.
Note: The harmonic sum is the sum of reciprocals of the positive integers.

Example:

$$1+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}+\frac{1}{5}+\cdots$$

For example:

Input	Result
5	2.283333333333333
7	2.5928571428571425

Answer: (penalty regime: 0 %)

```
def sum(n):
    i = 1
    s = 0.0
    for i in range(1, n+1):
        s = s + 1/i;
    return s;

# Driver Code
n = int(input())
print(sum(n))
```

	Input	Expected	Got	
✔	5	2.283333333333333	2.283333333333333	✔
✔	7	2.5928571428571425	2.5928571428571425	✔
✔	4	2.083333333333333	2.083333333333333	✔
✔	6	2.4499999999999997	2.4499999999999997	✔

Passed all tests! ✔

Marks for this submission: 2.00/2.00.

Question **5**

Correct

Mark 2.00 out of 2.00

Write a python program to print the following pattern

5 4 3 2 1
5 4 3 2
5 4 3
5 4
5

For example:

Input	Result
5	5 4 3 2 1 5 4 3 2 5 4 3 5 4 5
6	6 5 4 3 2 1 6 5 4 3 2 6 5 4 3 6 5 4 6 5 6

Answer: (penalty regime: 0 %)

```
def print_pattern(n):
    for i in range(n, 0, -1):
        for j in range(n, n-i, -1):
            print(j, end=" ")
        print()


n = int(input())
print_pattern(n)

```

	Input	Expected	Got	
✔	5	5 4 3 2 1 5 4 3 2 5 4 3 5 4 5	5 4 3 2 1 5 4 3 2 5 4 3 5 4 5	✔
✔	6	6 5 4 3 2 1 6 5 4 3 2 6 5 4 3 6 5 4 6 5 6	6 5 4 3 2 1 6 5 4 3 2 6 5 4 3 6 5 4 6 5 6	✔

	Input	Expected	Got	
✔	4	4 3 2 1 4 3 2 4 3 4	4 3 2 1 4 3 2 4 3 4	✔

Passed all tests! ✔

Correct

Marks for this submission: 2.00/2.00.