

CODETANTRA Home 202401070030@mitaoe.ac.in Support Logout

1.2.2. Fibonacci series using Recursive Function

Write a Python program to find the Fibonacci series of a given number of terms using recursive function calls.

Expected Output-1:
Enter terms for Fibonacci series: 5
0 1 1 2 3

Expected Output-2:
Enter terms for Fibonacci series: 9
0 1 1 2 3 5 8 13 21

Instructions:

- Your input and output must follow the input and output layout mentioned in the visible sample test case.
- Hidden test cases will only pass when users' input and output match the expected input and output.

fib.py

```
1 v def fib(n):
2 v     if(n <= 1):
3 v         return n
4 v     else:
5 v         return (fib(n-1) + fib(n-2))
6 n=int(input("Enter terms for Fibonacci series: "))
7 v for i in range (n):
8 v     print(fib(i),end=" ")
```

Sample Test Cases +

Terminal Test cases

CODETANTRA [Home](#)

202401070030@mitaoe.ac.in Support [Logout](#)

1.2.4. Pattern - 2

08:15

Write a Python program to print a right-angled triangle pattern of numbers.

Input Format:
The input is an integer, representing the number of rows in the pattern.

Output Format:
The output should display the pattern of numbers, with each row containing increasing numbers starting from 1 up to the row number.

Note:
Refer to the displayed test cases for the sample pattern.

Sample Test Cases

Explorer numberP...
1 n = int(input())
2 v for i in range(1,n+1):
3 v —— for j in range(i):
4 v —— —— print(j+1,end = " ")
5 —— print("")

Terminal Test cases

[« Prev](#) [Reset](#) [Submit](#) [Next »](#)

CODETANTRA Home

202401070030@mitaoe.ac.in Support Logout

3.2.3. Numpy: Custom Sequence Generation

Write a Python program that takes the following inputs from the user:

- Start value: The starting point of the sequence.
- Stop value: The sequence should end before this value.
- Step value: The increment between each number in the sequence.

The program should then generate a sequence using `numpy` based on these inputs and print the generated sequence.

Input Format:

- The user will input three integer values: start, stop, and step, each on a new line.

Output Format:

- The program should print the generated sequence based on the input values.

customS...

```
1 import numpy as np
2
3 # Take user input for the start, stop, and step of the sequence
4 start = int(input())
5 stop = int(input())
6 step = int(input())
7
8 # Generate the sequence using np.arange()
9 p=np.arange(start,stop,step)
10 # Print the generated sequence
11 print(p)
12
```

Sample Test Cases +

Terminal Test cases

< Prev Reset Submit Next >

CODETANTRA Home

202401070030@mitaoe.ac.in Support Logout

3.2.5. Numpy: Copying and Viewing Arrays

The given code takes a list of integers as input and converts it into a NumPy array. Your task is to complete the code by:

- Creating a view of the `original_array` and assigning it to `view_array`.
- Creating a copy of the `original_array` and assigning it to `copy_array`.

After completing these steps, observe how modifying the view affects the `original_array`, while modifying the copy does not.

Input Format:

- A single line of space-separated integers.

Output Format:

- After modifying the view:

```
Original array after modifying view: <original_array>
View array: <view_array>
```

- After modifying the copy:

```
Original array after modifying copy: <original_array>
Copy array: <copy_array>
```

Sample Test Cases +

copyAnd... Submit

```
import numpy as np
inputlist = list(map(int,input().split(" ")))
# Original array
original_array = np.array(inputlist)
# Create a view
view_array = original_array.view()
# Create a copy
copy_array = original_array.copy()
# Modify the view
view_array[0] = 99
print("Original array after modifying view:", original_array)
print("View array:", view_array)
# Modify the copy
copy_array[1] = 88
print("Original array after modifying copy:", original_array)
print("Copy array:", copy_array)
```

Terminal Test cases < Prev Reset Next >

CODETANTRA Home

202401070030@mitaoe.ac.in Support Logout

3.2.6. Numpy: Searching, Sorting, Counting, Broadcasting

The given code in the editor takes a single array, `array1`, as space-separated integers as input from the user.

Additionally, it takes the following inputs:

- `search_value`: The value to search for in the array.
- `count_value`: The value to count its occurrences in the array.
- `broadcast_value`: The value to add for broadcasting across the array.

You need to complete the code to perform the following operations:

1. **Searching**: Find the indices where `search_value` appears in `array1` and print these indices.
2. **Counting**: Count how many times `count_value` appears in `array1` and print the count.
3. **Broadcasting**: Add `broadcast_value` to each element of `array1` using broadcasting, and print the resulting array.
4. **Sorting**: Sort `array1` in ascending order and print the sorted array.

Input Format:

1. A single line containing space-separated integers representing `array1`.
2. An integer `search_value` represents the value to search for in the array.
3. An integer `count_value` represents the value to count in the array.
4. An integer `broadcast_value` represents the value to add to each element of the array.

Sample Test Cases +

arrayOpe...
import numpy as np
Input array from the user
array1 = np.array(list(map(int, input().split())))

Searching
search_value = int(input("Value to search: "))
count_value = int(input("Value to count: "))
broadcast_value = int(input("Value to add: "))

Find indices where value matches in array1
search=np.where(array1==search_value)
print(search[0])
Count occurrences in array1
count=np.count_nonzero(array1==count_value)
print(count)
Broadcasting addition
broadcast_result=array1+broadcast_value
print(broadcast_result)
Sort the first array
sorted_array=np.sort(array1)
print(sorted_array)

Terminal Test cases < Prev Reset Submit Next >

CODETANTRA Home 202401070030@mitaoe.ac.in Support Logout

3.2.1. Numpy: Matrix Operations 13:20

The given code takes two 3×3 matrices, `matrix_a`, and `matrix_b`, as input from the user and converts them into NumPy arrays.

Task:
You are required to compute and display the results of the following matrix operations:
1. Addition (`matrix_a + matrix_b`)
2. Subtraction (`matrix_a - matrix_b`)
3. Element-wise Multiplication (`matrix_a * matrix_b`)
4. Matrix Multiplication (`matrix_a . matrix_b`)
5. Transpose of Matrix A

Input Format:
The user will input 3 rows for `matrix_a`, each containing 3 integers separated by spaces.
Similarly, the user will input 3 rows for `matrix_b`, each containing 3 integers separated by spaces.

Output Format:
The program should display the results of the operations in the following order:
1. The result of Addition.
2. The result of Subtraction.

Sample Test Cases +

```
matrixOp...
5  matrix_a = np.array([list(map(int, input().split())) for i in range(3)])
6
7  print("Enter Matrix B:")
8  matrix_b = np.array([list(map(int, input().split())) for i in range(3)])
9  addition=np.add(matrix_a,matrix_b)
10
11 # Addition
12 print("Addition (A + B):")
13 print(addition)
14 # Subtraction
15 subtract=np.subtract(matrix_a,matrix_b)
16 print("Subtraction (A - B):")
17 print(subtract)
18 # Multiplication (element-wise)
19 multiplicatin=np.multiply(matrix_a,matrix_b)
20 print("Element-wise Multiplication (A * B):")
21 print(multiplicatin)
22 # Matrix multiplication (dot product)
23 Matrix_Multiplicatin=np.dot(matrix_a,matrix_b)
24 print("A dot B:")
25 print(Matrix_Multiplicatin)
26 # Transpose
27 Transpose=np.transpose(matrix_a)
28 print("Transpose of A:")
29 print(Transpose)
```

< Prev Reset Submit Next >

CODETANTRA Home 202401070030@mitaoe.ac.in Support Logout

3.2.1. Numpy: Matrix Operations 13:20

The given code takes two 3×3 matrices, `matrix_a`, and `matrix_b`, as input from the user and converts them into NumPy arrays.

Task:
You are required to compute and display the results of the following matrix operations:
1. Addition (`matrix_a + matrix_b`)
2. Subtraction (`matrix_a - matrix_b`)
3. Element-wise Multiplication (`matrix_a * matrix_b`)
4. Matrix Multiplication (`matrix_a . matrix_b`)
5. Transpose of Matrix A

Input Format:
The user will input 3 rows for `matrix_a`, each containing 3 integers separated by spaces.
Similarly, the user will input 3 rows for `matrix_b`, each containing 3 integers separated by spaces.

Output Format:
The program should display the results of the operations in the following order:
1. The result of Addition.
2. The result of Subtraction.

Sample Test Cases +

```
matrixOp...
5  matrix_a = np.array([list(map(int, input().split())) for i in range(3)])
6
7  print("Enter Matrix B:")
8  matrix_b = np.array([list(map(int, input().split())) for i in range(3)])
9  addition=np.add(matrix_a,matrix_b)
10
11 # Addition
12 print("Addition (A + B):")
13 print(addition)
14 # Subtraction
15 subtract=np.subtract(matrix_a,matrix_b)
16 print("Subtraction (A - B):")
17 print(subtract)
18 # Multiplication (element-wise)
19 multiplicatin=np.multiply(matrix_a,matrix_b)
20 print("Element-wise Multiplication (A * B):")
21 print(multiplicatin)
22 # Matrix multiplication (dot product)
23 Matrix_Multiplicatin=np.dot(matrix_a,matrix_b)
24 print("A dot B:")
25 print(Matrix_Multiplicatin)
26 # Transpose
27 Transpose=np.transpose(matrix_a)
28 print("Transpose of A:")
29 print(Transpose)
```

< Prev Reset Submit Next >

CodeTANTRA Home

202401070030@mitaoe.ac.in Support Logout

1.2.3. Pattern - 1 02:31

Write a Python program to print a pattern of asterisks in the form of a right-angled triangle.

Input Format:
The input is an integer, representing the number of rows in the pattern.

Output Format
The output should display the pattern of asterisks (*), with each row containing an increasing number of asterisks.

Note:
Refer to the displayed test cases for the sample pattern.

Sample Test Cases +

rightangl... Explorer

```
1 n = int(input())
2 v for i in range(1,n+1):
3 v   for j in range(0,i):
4     print("*",end = " ")
5   print("")
```

Submit Debugger

Terminal Test cases

< Prev Reset Submit Next >