

< Prev

Reset

Submit





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.

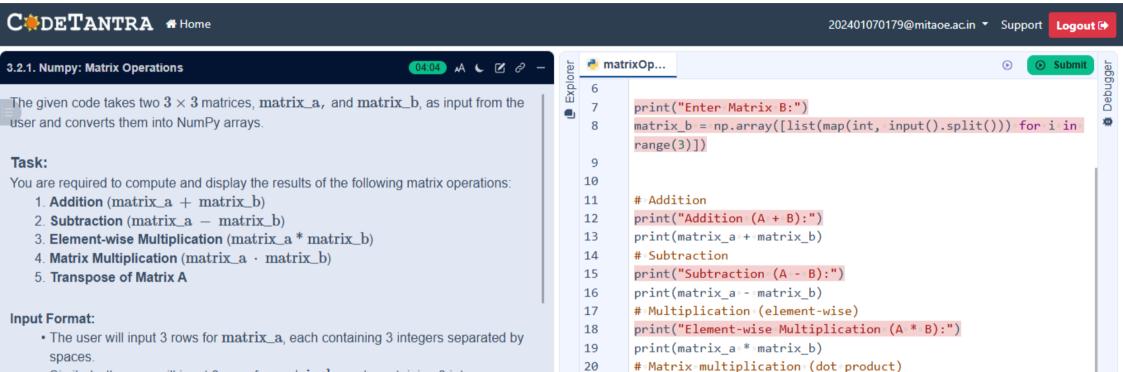
Sample Test Cases

Explorer matrixOp... Submit Debugger 1 import numpy as np 2 3 # Input matrices print("Enter Matrix A:") 4 matrix_a = np.array([list(map(int, input().split())) for i in 5 range(3)]) 6 print("Enter Matrix B:") 7 matrix b = np.array([list(map(int, input().split()))) for i in 8 range(3)]) 9 10 11 # Addition print("Addition (A + B):") 12 print(matrix a + matrix b) 13 # Subtraction 14 print("Subtraction (A - B):") 15 print(matrix a - matrix b) 16 # Multiplication (element-wise) 17 print("Element-wise Multiplication (A * B):") 18 print(matrix a * matrix b) 19 Activate Windows > Terminal Test cases

< Prev

Reset

Submit



21

22

23

24

25

>_ Terminal

print("A dot B:")

print(matrix_a.T)

print("Transpose of A:")

 ☐ Test cases

Transpose

print(np.dot(matrix a,matrix b))

• Similarly, the user will input 3 rows for matrix_b, each containing 3 integers

separated by spaces.

Sample Test Cases

< Prev

Activate Windows

Reset

Submit

Debugger



You are given two arrays arr1 and arr2. You need to perform horizontal and vertical stacking operations on them using NumPy.

- · Horizontal Stacking: Stack the two matrices horizontally (side by side).
- Vertical Stacking: Stack the two matrices vertically (one below the other).

Input Format:

- The program should first prompt the user to input two 3x3 arrays.
- Each array consists of 3 rows, and each row contains 3 space-separated integers.
- . The user will input the two arrays row by row.

Output Format:

- The program should display the result of the Horizontal Stack (side-by-side stacking) of the two arrays.
- The program should then display the result of the Vertical Stack (one below the other) of the two arrays.

Sample Test Cases

12 13 14 15

import numpy as np

stacking.py

Explorer 1

2

16 17

print("Vertical Stack:") 18

>_ Terminal ⊞ Test cases Activate Windows

< Prev

Reset

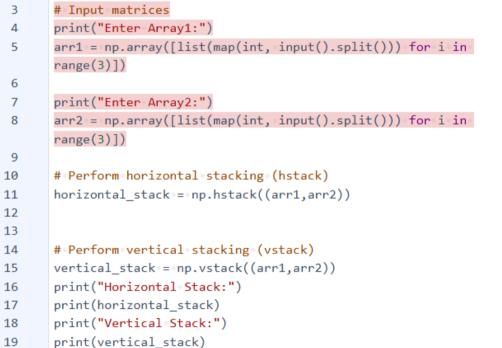
Submit

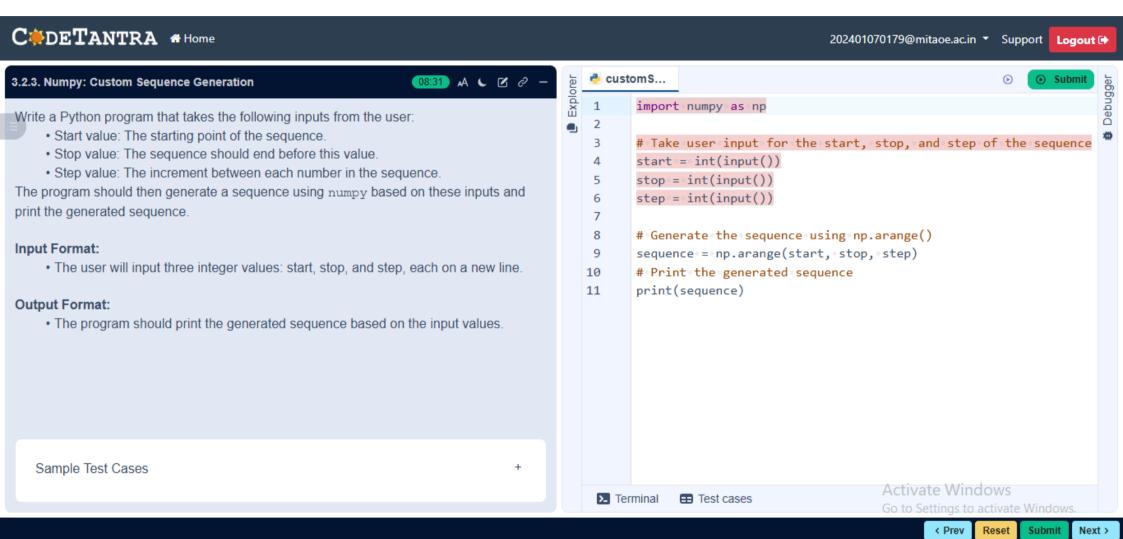




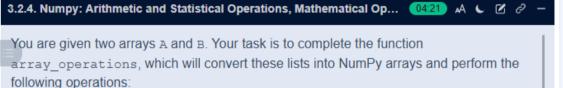












1. Arithmetic Operations:

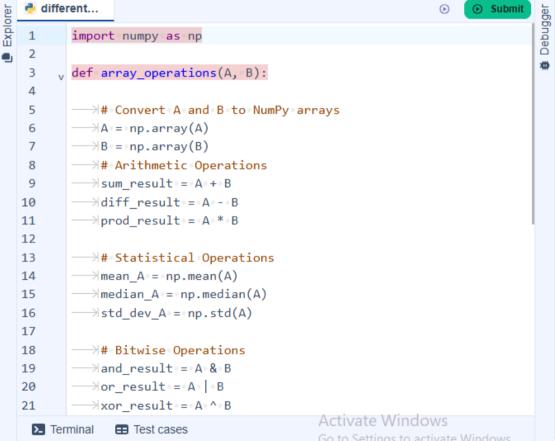
- · Compute the element-wise sum, difference, and product of the two arrays.
- 2. Statistical Operations:
 - · Calculate the mean, median, and standard deviation of array A.
- 3. Bitwise Operations:
 - Perform bitwise AND, bitwise OR, and bitwise XOR on the arrays (ex: A₁ OR B₁).

Input Format:

- The first line contains space-separated integers representing the elements of array
- The second line contains space-separated integers representing the elements of array B.

Sample Test Cases

different...











Debugger



You are given two arrays A and B. Your task is to complete the function array operations, which will convert these lists into NumPy arrays and perform the following operations:

1. Arithmetic Operations:

- Compute the element-wise sum, difference, and product of the two arrays.
- 2. Statistical Operations
 - Calculate the mean, median, and standard deviation of array A.
- 3. Bitwise Operations:
 - Perform bitwise AND, bitwise OR, and bitwise XOR on the arrays (ex: A₁ OR B₁).

Input Format:

- The first line contains space-separated integers representing the elements of array
- The second line contains space-separated integers representing the elements of array B.

Sample Test Cases

Explorer 20 different... Submit and result = A & B or result = A B 21 xor result = A ^ B 22 23 # Output results with one space between each element print("Element-wise Sum:", ' '.join(map(str, sum_result))) 24 25 print("Element-wise Difference:", ''.join(map(str, diff result))) print("Element-wise Product:", ' '.join(map(str, 26 prod result))) 27 print(f"Mean of A: {mean A}") 28 print(f"Median of A: {median A}") 29 print(f"Standard Deviation of A: {std dev A}") 30 31 print("Bitwise AND:", ' '.join(map(str, and_result))) 32 print("Bitwise OR:", ' '.join(map(str, or_result))) 33 print("Bitwise XOR:", ' '.join(map(str, xor result))) 34 35 A = list(map(int, input().split())) # Elements of array A 36 B = list(map(int, input().split())) # Elements of array B 37 array operations(A R) Activate Windows ⊞ Test cases > Terminal

< Prev

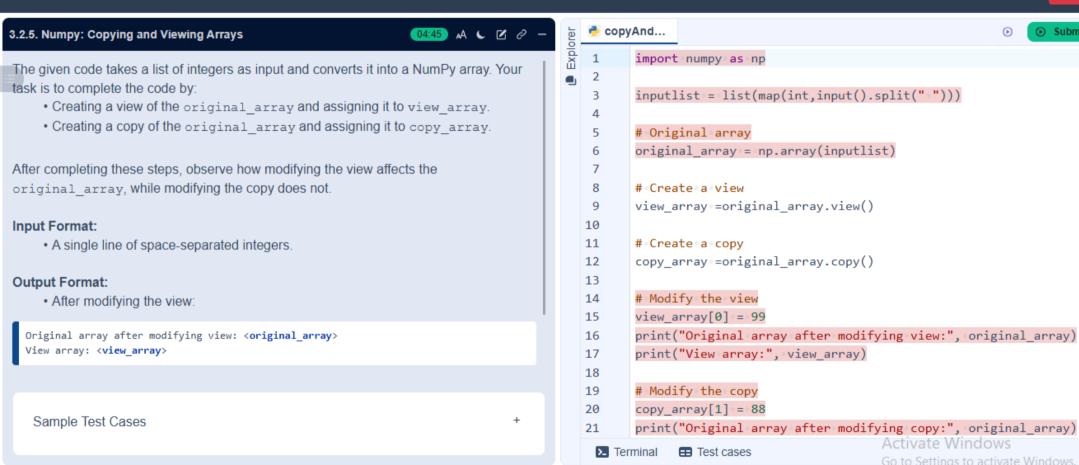
Reset

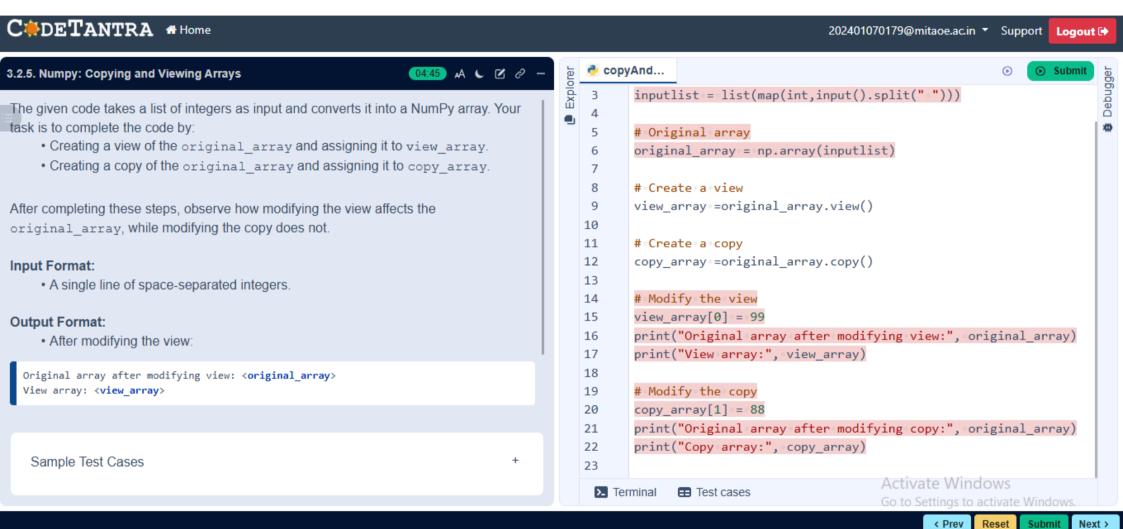
Submit

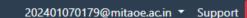
Submit

Debugger















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.

Innut Format:

Sample Test Cases

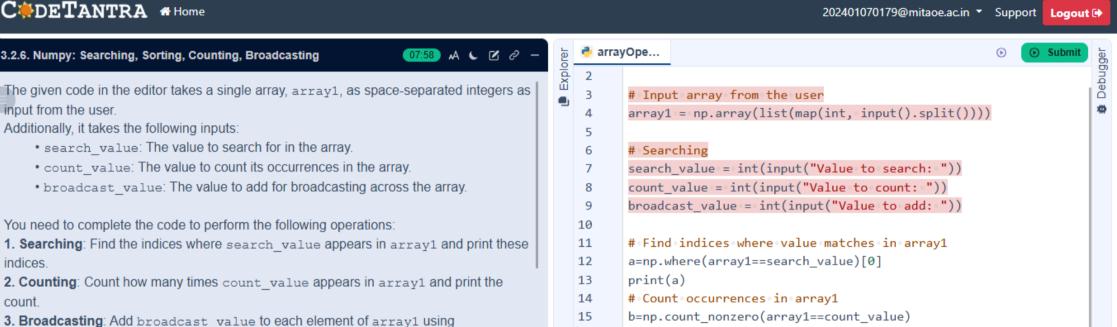
Debugger arrayOpe... Explorer 1 import numpy as np 2 # Input array from the user 3 array1 = np.array(list(map(int, input().split()))) 4 5 6 # Searching 7 search value = int(input("Value to search: ")) count value = int(input("Value to count: ")) 8 broadcast value = int(input("Value to add: ")) 9 10 11 # Find indices where value matches in arrav1 12 a=np.where(array1==search value)[0] 13 print(a) 14 # Count occurrences in array1 b=np.count_nonzero(array1==count_value) 15 print(b) 16 # Broadcasting addition 17 18 c=array1+broadcast value 19 print(c) 20 # Sort the first array d=np.sort(array1) 21 Activate Windows >_ Terminal ☐ Test cases





Submit





Broadcasting addition

c=array1+broadcast value

Sort the first array d=np.sort(array1)

 ☐ Test cases

print(b)

print(c)

print(d)

16

17

18

19

20

21

22

> Terminal

Innut Format:

indices

count.

Sample Test Cases

CODETANTRA # Home

Additionally, it takes the following inputs:

broadcasting, and print the resulting array.

4. Sorting: Sort array1 in ascending order and print the sorted array.

input from the user.



Activate Windows

< Prev

Reset

Submit

Next >

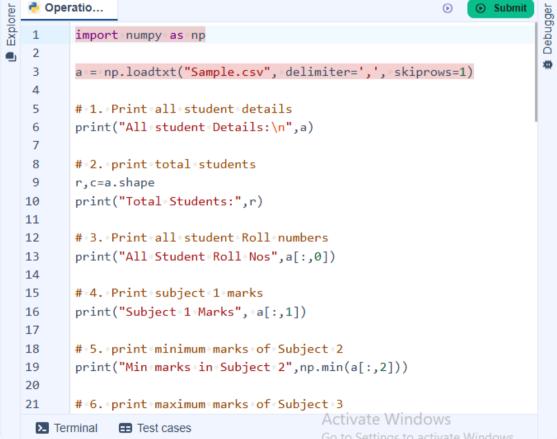


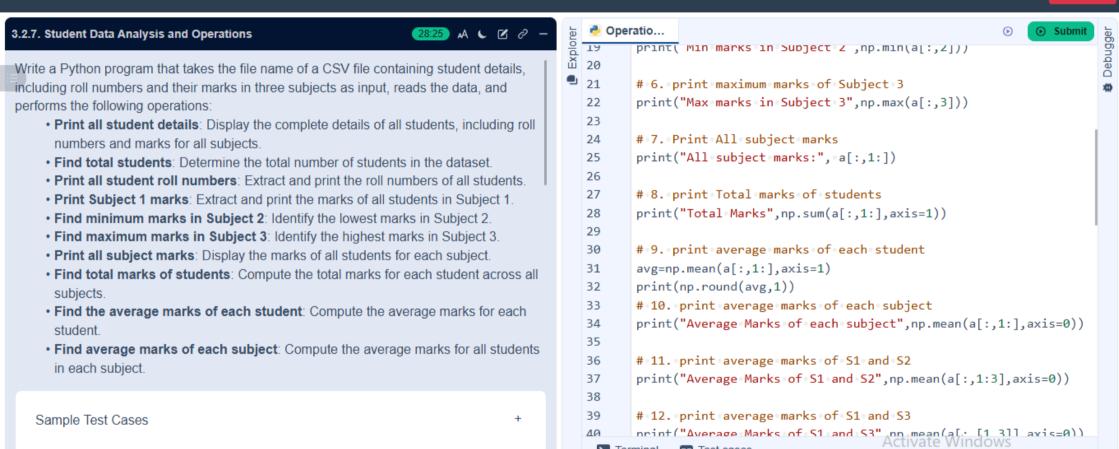


Write a Python program that takes the file name of a CSV file containing student details. including roll numbers and their marks in three subjects as input, reads the data, and performs the following operations:

- Print all student details: Display the complete details of all students, including roll numbers and marks for all subjects.
- Find total students: Determine the total number of students in the dataset.
- Print all student roll numbers: Extract and print the roll numbers of all students.
- Print Subject 1 marks: Extract and print the marks of all students in Subject 1.
- Find minimum marks in Subject 2: Identify the lowest marks in Subject 2.
- Find maximum marks in Subject 3: Identify the highest marks in Subject 3.
- · Print all subject marks: Display the marks of all students for each subject.
- Find total marks of students: Compute the total marks for each student across all subjects.
- Find the average marks of each student: Compute the average marks for each student.
- Find average marks of each subject: Compute the average marks for all students in each subject.

Sample Test Cases





>_ Terminal

⊞ Test cases





Write a Python program that takes the file name of a CSV file containing student details. including roll numbers and their marks in three subjects as input, reads the data, and performs the following operations:

- Print all student details: Display the complete details of all students, including roll numbers and marks for all subjects.
- Find total students: Determine the total number of students in the dataset.
- Print all student roll numbers: Extract and print the roll numbers of all students.
- Print Subject 1 marks: Extract and print the marks of all students in Subject 1.
- Find minimum marks in Subject 2: Identify the lowest marks in Subject 2.
- Find maximum marks in Subject 3: Identify the highest marks in Subject 3.
- Print all subject marks: Display the marks of all students for each subject.
- Find total marks of students: Compute the total marks for each student across all subjects.
- Find the average marks of each student: Compute the average marks for each student
- Find average marks of each subject: Compute the average marks for all students in each subject.

Sample Test Cases

#1 42 Debugger Operatio... Submit # 13. print Roll number who got maximum marks in Subject 3 **4**3 i=np.argmax(a[:,3])print("Roll no who got maximum marks in Subject 3",a[i,0]) 44 45 # 14. print Roll number who got minimum marks in Subject 2 46 47 mn=np.argmin(a[:,2])48 print("Roll no who got minimum marks in Subject 2",a[mn,0]) 49 50 # 15. print Roll number who got 24 marks in Subject 2 whr=np.where(a[:,2]==24) 51 52 print("Roll no who got 24 marks in Subject 2",a[whr,0]) 53 54 # 16. print count of students who got marks in Subject 1 < 40 55 ct=np.count nonzero(a[:,1]<40) print("Count of students who got marks in Subject 1 < 40",ct)</pre> 56 57 58 # 17. print count of students who got marks in Subject 2 > 90 59 ct1=np.count nonzero(a[:,2]>90) print("Count of students who got marks in Subject 2 > 90:",ct1) 60 61 # 18. print count of students in each subject who got marks >= Activate Windows 62 >_ Terminal ⊞ Test cases

< Prev

Reset

Submit







Write a Python program that takes the file name of a CSV file containing student details. including roll numbers and their marks in three subjects as input, reads the data, and performs the following operations:

- Print all student details: Display the complete details of all students, including roll numbers and marks for all subjects.
- Find total students: Determine the total number of students in the dataset.
- Print all student roll numbers: Extract and print the roll numbers of all students.
- Print Subject 1 marks: Extract and print the marks of all students in Subject 1.
- Find minimum marks in Subject 2: Identify the lowest marks in Subject 2.
- Find maximum marks in Subject 3: Identify the highest marks in Subject 3:
- · Print all subject marks: Display the marks of all students for each subject.
- · Find total marks of students: Compute the total marks for each student across all subjects.
- Find the average marks of each student: Compute the average marks for each student
- · Find average marks of each subject: Compute the average marks for all students in each subject.

Sample Test Cases

Explorer 61 Operatio... Debugger 62 # 18. print count of students in each subject who got marks >= 90 print("Count of students in each subject who got marks >= 63 90:",np.count nonzero(a[:,1:]>=90,axis=0)) 64 # 19. print count of subjects in which each student got marks 65 >= 90 print("Roll no:",a[:,0]) 66 print("Count of subjects in which student got marks >= 67 90:",np.count nonzero(a[:,1:]>=90,axis=1)) 68 69 # 20. Print S1 marks in ascending order 70 srt=np.sort(a[:,1]) 71 print(srt) 72 73 # 21. Print S1 marks >= 50 and <= 90 74 print(a[(a[:,1]>=50)&(a[:,1]<90)]) 75 print(a) 76 ip=np.where(a[:,1]==79) print(ip) 77 Activate Windows >_ Terminal ⊞ Test cases