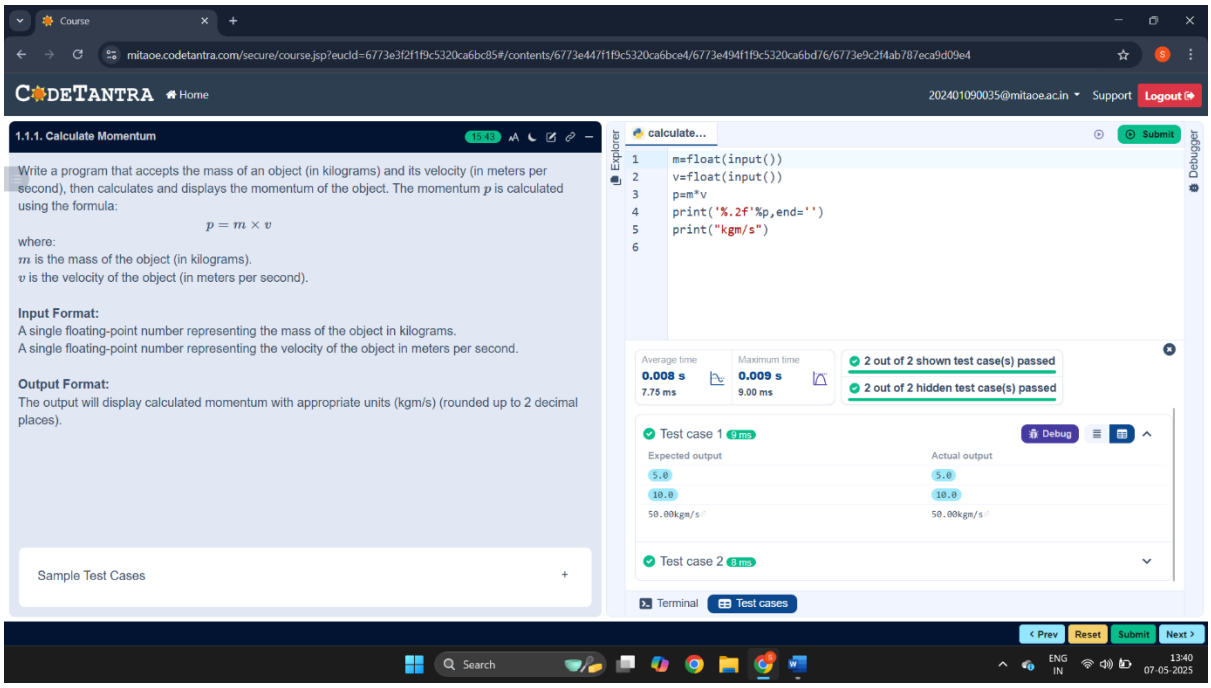
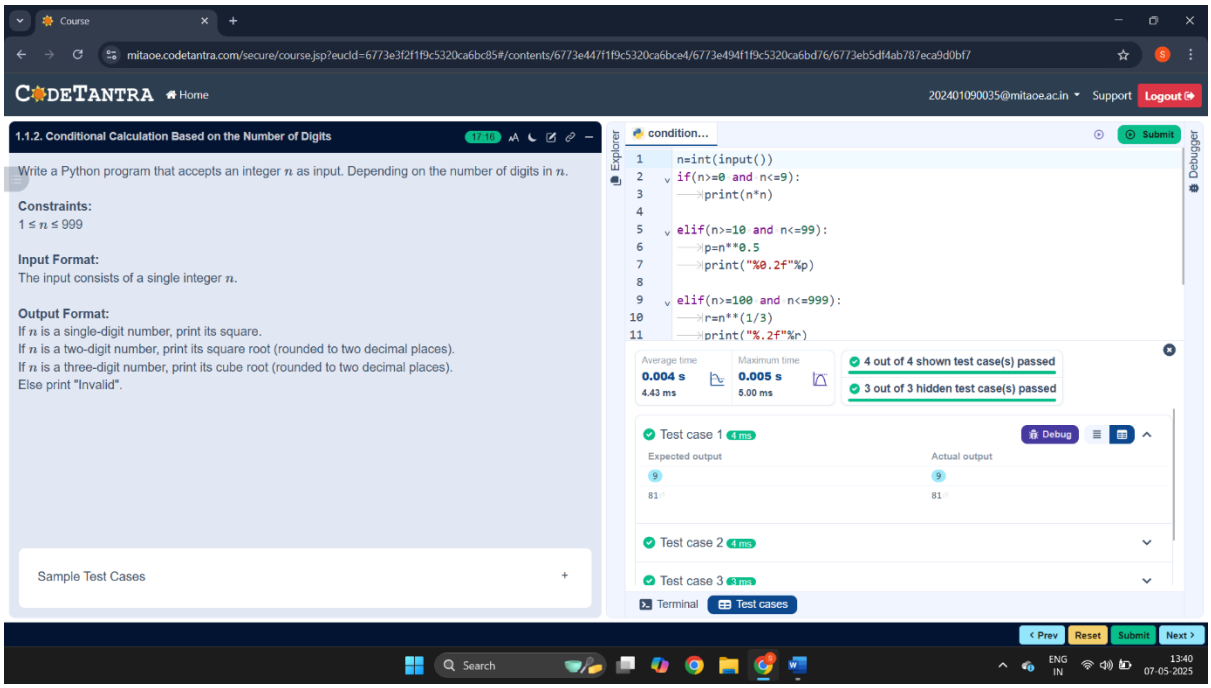
Name :- Shraddha Pravin Sonawane.

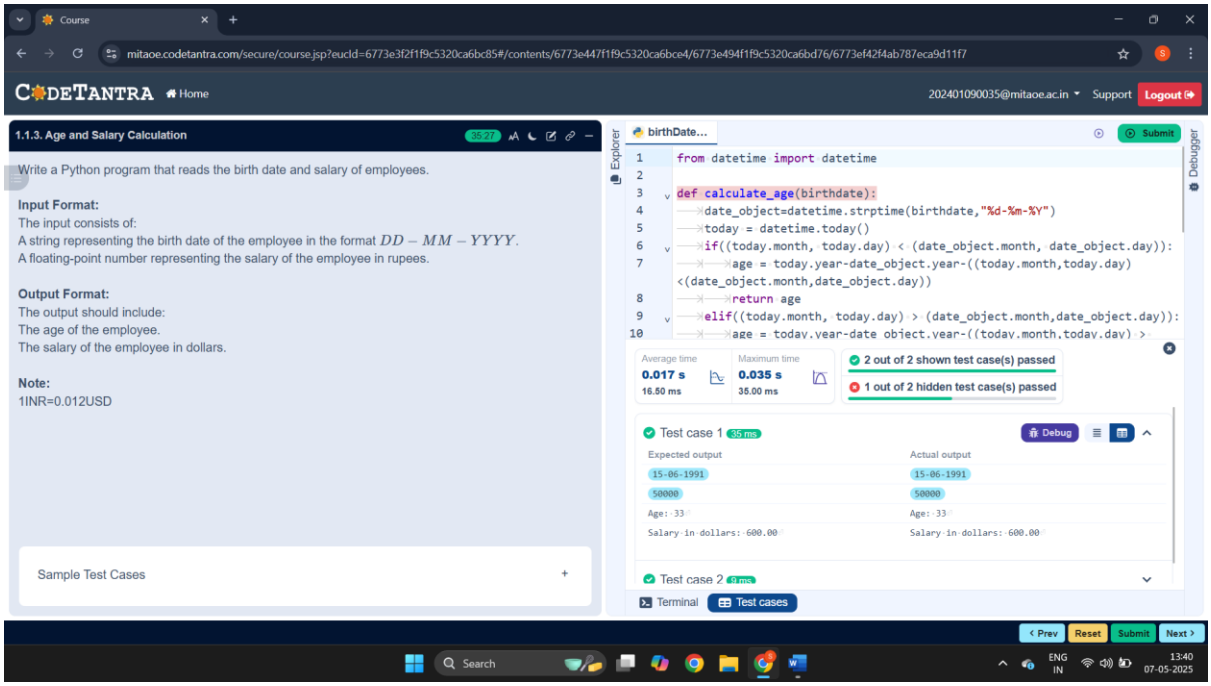
Batch:- ME2

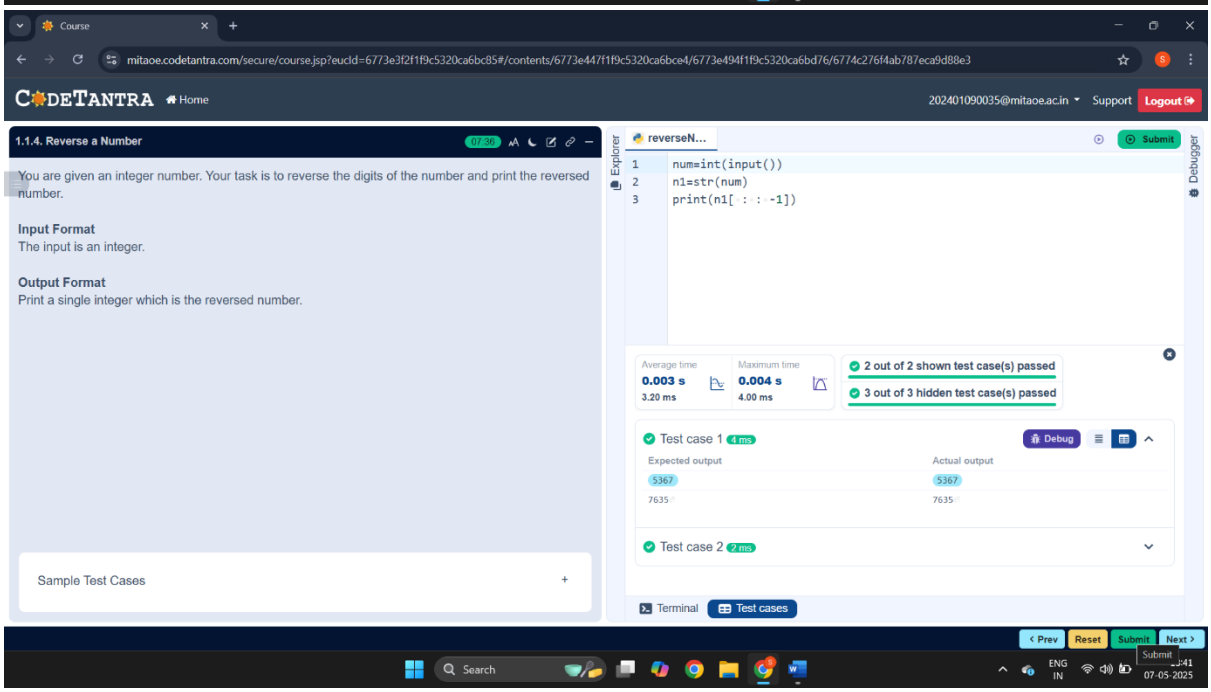
Roll no. :- ME33

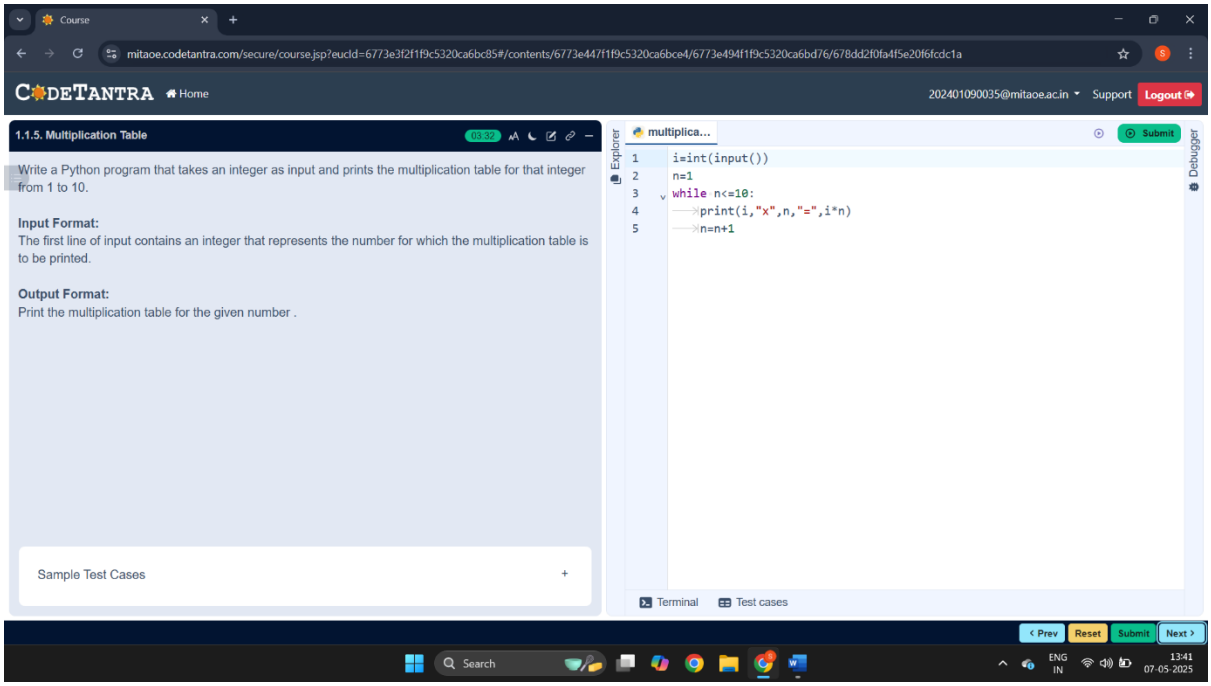
Practical no. 1

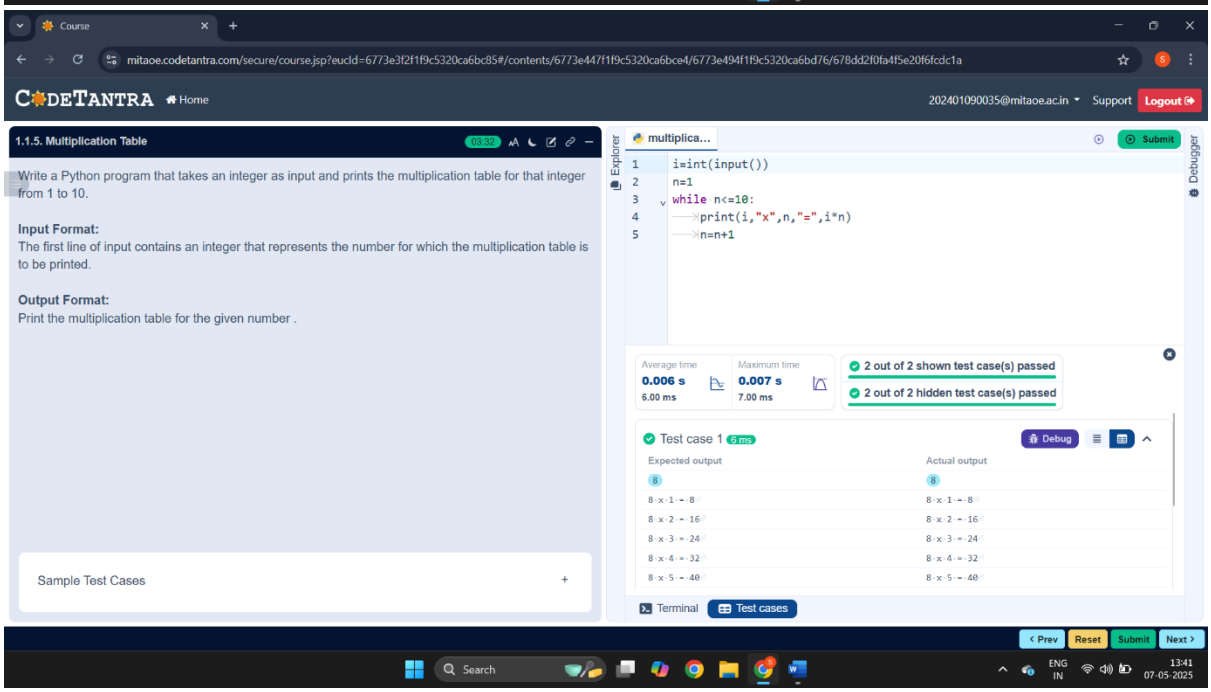


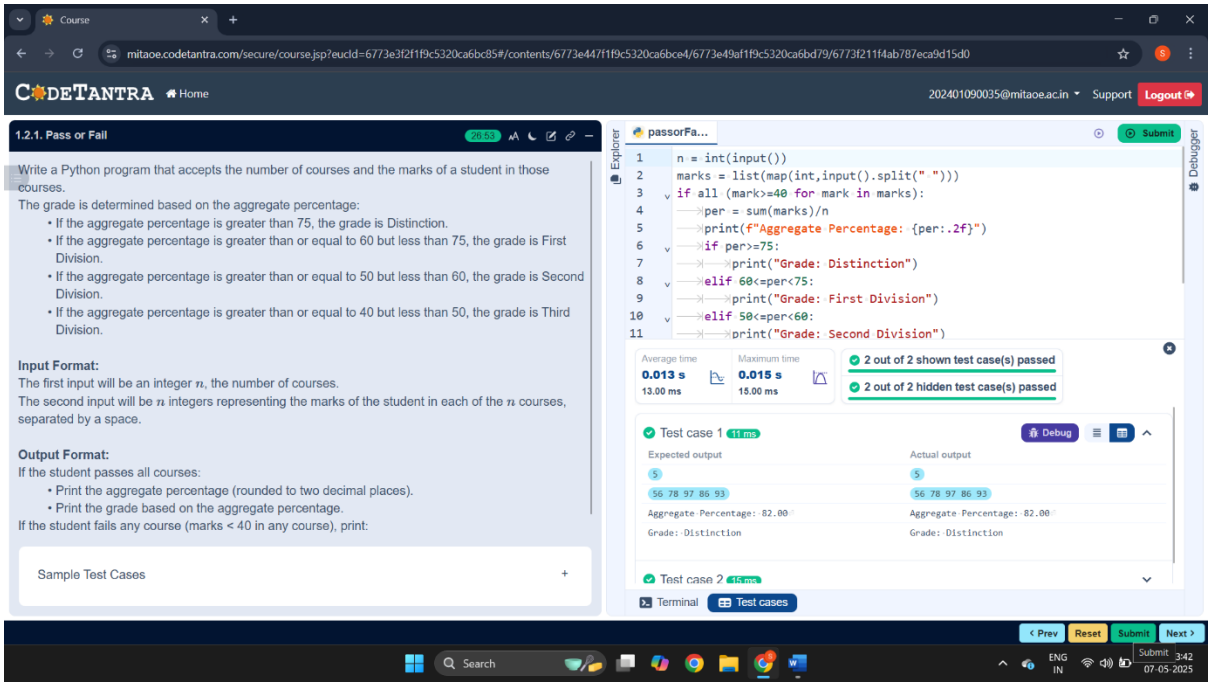


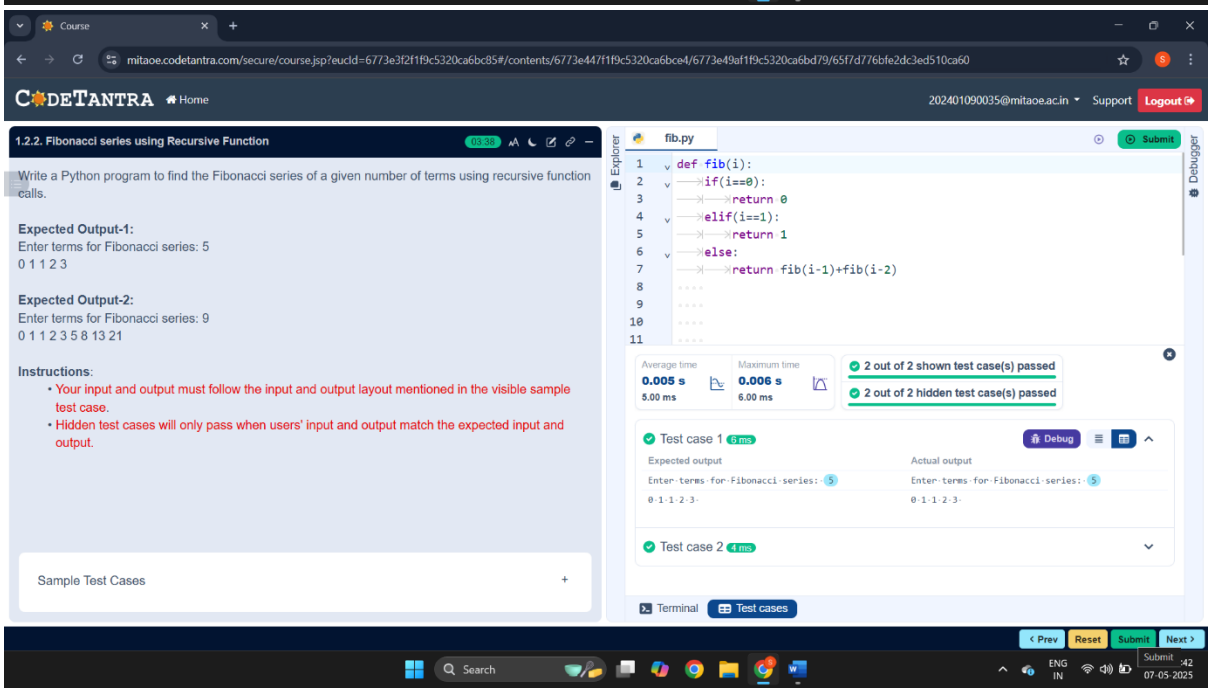

Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e447f1f9c5320ca6bce4/6773e494f1f9c5320ca6bd76/6773ef42f4ab787eca9d11f7
CODETANTRA
Home
202401090035@mitaoe.ac.in
Support
Logout
1.1.3. Age and Salary Calculation
35:27
Write a Python program that reads the birth date and salary of employees.
Input Format:
The input consists of:
A string representing the birth date of the employee in the format $DD-MM-YYYY$.
A floating-point number representing the salary of the employee in rupees.
Output Format:
The output should include:
The age of the employee.
The salary of the employee in dollars.
Note:
1INR=0.012USD
Sample Test Cases
birthDate...
Submit
Explorer
Debugger
from datetime import datetime
def calculate_age(birthdate):
date_object=datetime.strptime(birthdate,"%d-%m-%Y")
today = datetime.today()
if((today.month, today.day) < (date_object.month, date_object.day)):
age = today.year-date_object.year-((today.month,today.day)
<(date_object.month,date_object.day))
return age
elif((today.month, today.day) > (date_object.month,date_object.day)):
age = today.year-date_object.year-((today.month,today.day) >
Average time
0.017 s
16.50 ms
Maximum time
0.035 s
35.00 ms
2 out of 2 shown test case(s) passed
1 out of 2 hidden test case(s) passed
Test case 1
35 ms
Debug
Expected output
15-06-1991
50000
Age: 33
Salary in dollars: 600.00
Actual output
15-06-1991
50000
Age: 33
Salary in dollars: 600.00
Test case 2
Terminal
Test cases
‹ Prev
Reset
Submit
Next ›
Search
ENG
IN
13:40
07-05-2025

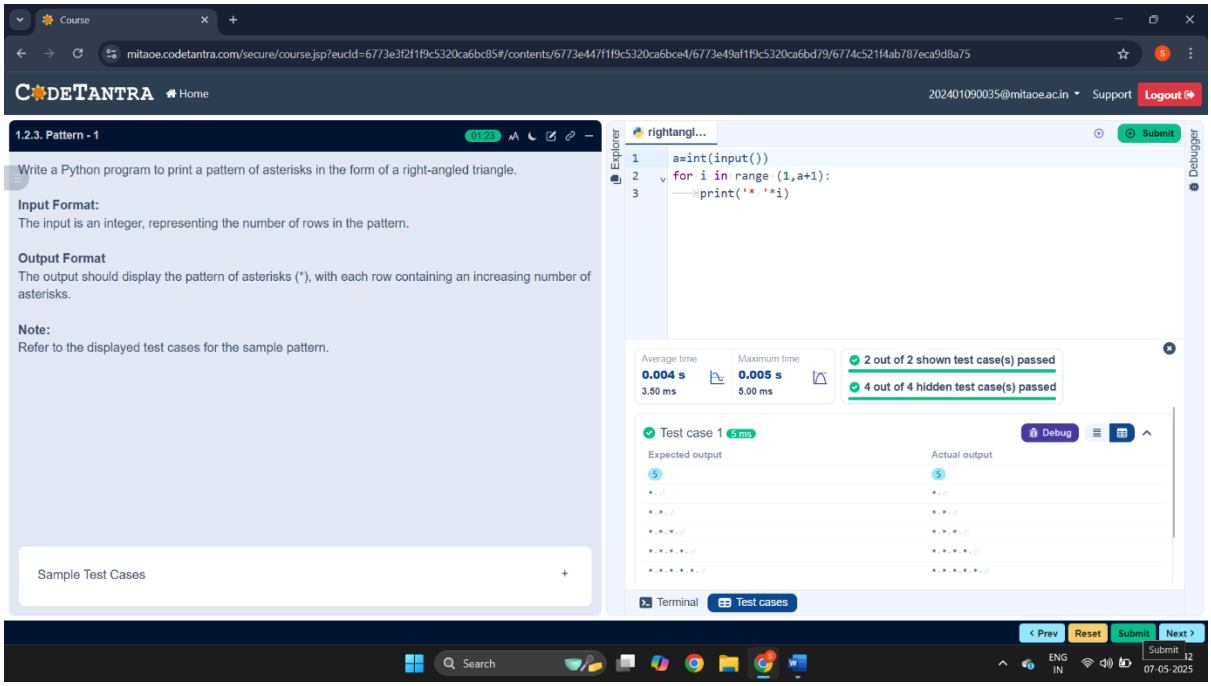

Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e447f1f9c5320ca6bce4/6773e494f1f9c5320ca6bd76/6774c276f4ab787eca9d88e3
CODETANTRA
Home
202401090035@mitaoe.ac.in
Support
Logout
1.1.4. Reverse a Number
07:36
You are given an integer number. Your task is to reverse the digits of the number and print the reversed number.
Input Format
The input is an integer.
Output Format
Print a single integer which is the reversed number.
Sample Test Cases
reverseN...
Submit
Explorer
Debugger
num=int(input())
n1=str(num)
print(n1[: : -1])
Average time
0.003 s
3.20 ms
Maximum time
0.004 s
4.00 ms
2 out of 2 shown test case(s) passed
3 out of 3 hidden test case(s) passed
Test case 1
4 ms
Debug
Expected output
5367
7635
Actual output
5367
7635
Test case 2
2 ms
Terminal
Test cases
‹ Prev
Reset
Submit
Next ›
Search
ENG
IN
07-05-2025

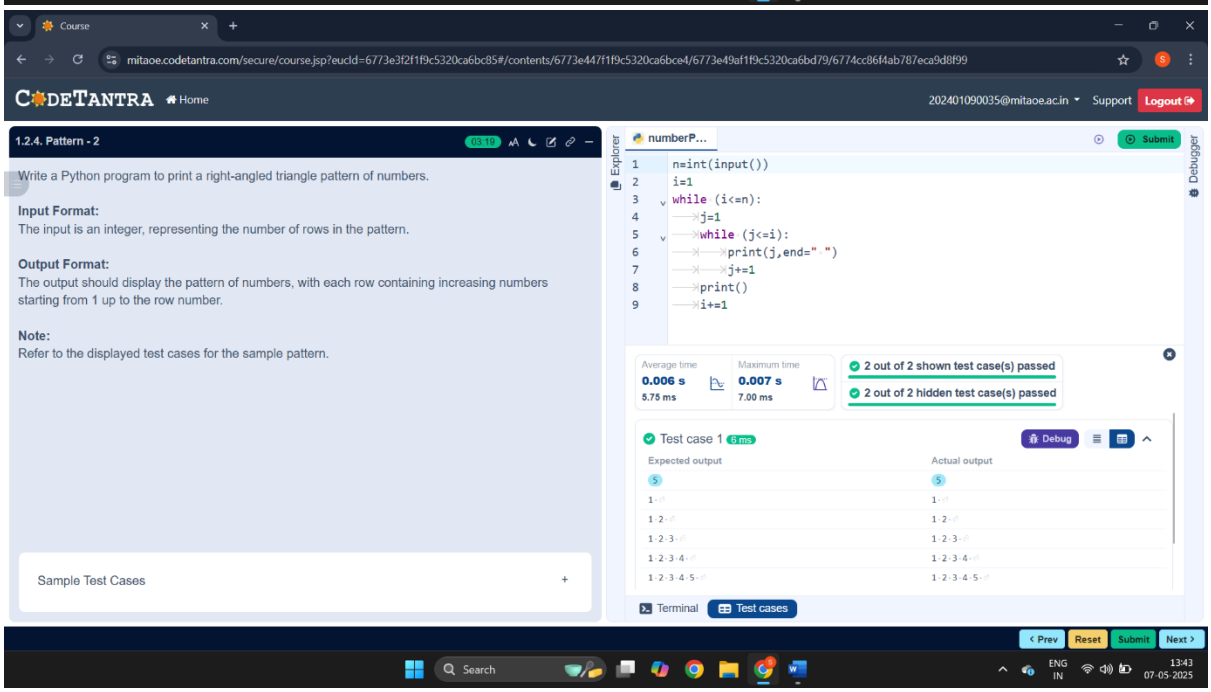

Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e447f1f9c5320ca6bce4/6773e494f1f9c5320ca6bd76/678dd2f0fa4f5e20f6fcdc1a
CODETANTRA
Home
202401090035@mitaoe.ac.in
Support
Logout
1.1.5. Multiplication Table
03:32
Write a Python program that takes an integer as input and prints the multiplication table for that integer from 1 to 10.
Input Format:
The first line of input contains an integer that represents the number for which the multiplication table is to be printed.
Output Format:
Print the multiplication table for the given number .
Sample Test Cases
multiplica...
Submit
Explorer
Debugger
1 i=int(input())
2 n=1
3 while n<=10:
4 print(i,"x",n,"=",i*n)
5 n=n+1
Terminal
Test cases
Prev
Reset
Submit
Next
Search
ENG
IN
13:41
07-05-2025


Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e447f1f9c5320ca6bce4/6773e494f1f9c5320ca6bd76/678dd2f0fa4f5e20f6fcdc1a
CODETANTRA
Home
202401090035@mitaoe.ac.in
Support
Logout
1.1.5. Multiplication Table
03:32
Write a Python program that takes an integer as input and prints the multiplication table for that integer from 1 to 10.
Input Format:
The first line of input contains an integer that represents the number for which the multiplication table is to be printed.
Output Format:
Print the multiplication table for the given number .
Sample Test Cases
multiplica...
Submit
Explorer
Debugger
1 i=int(input())
2 n=1
3 while n<=10:
4 print(i,"x",n,"=",i*n)
5 n=n+1
Average time
0.006 s
6.00 ms
Maximum time
0.007 s
7.00 ms
2 out of 2 shown test case(s) passed
2 out of 2 hidden test case(s) passed
Test case 1
6 ms
Debug
Expected output
Actual output
8
8 x 1 = 8
8 x 2 = 16
8 x 3 = 24
8 x 4 = 32
8 x 5 = 40
Terminal
Test cases
Prev
Reset
Submit
Next
Search
ENG
IN
13:41
07-05-2025

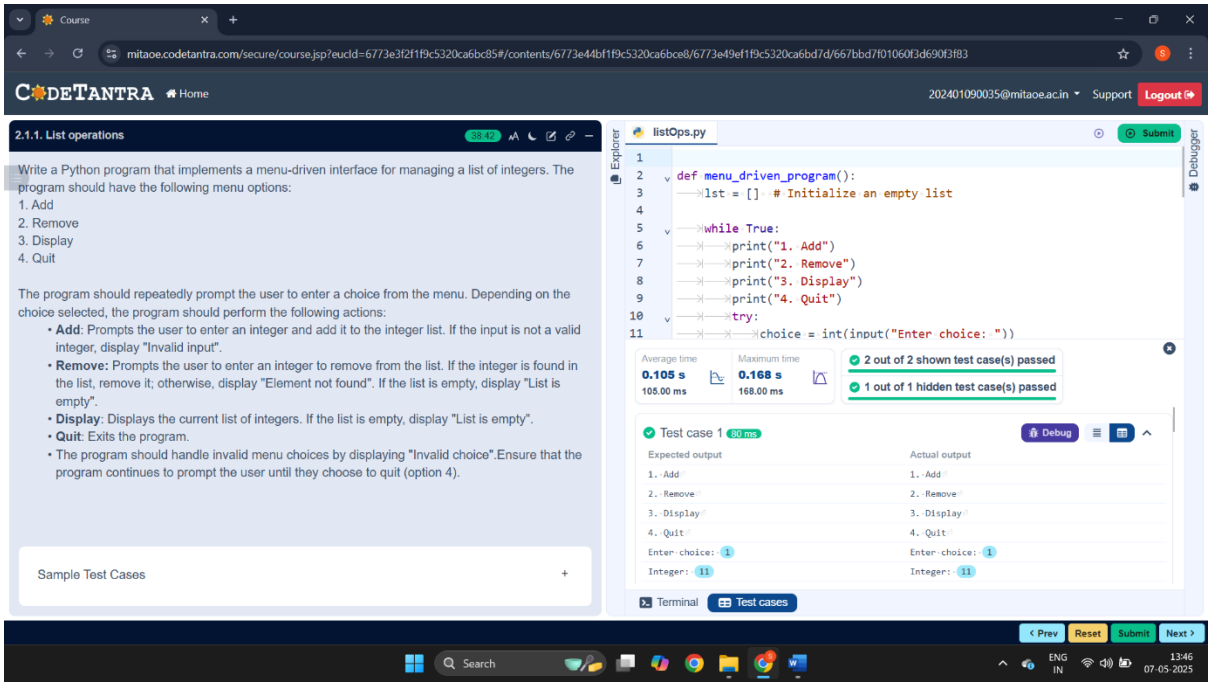

Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e447f1f9c5320ca6bce4/6773e49af1f9c5320ca6bd79/6773f211f4ab787eca9d15d0
CODETANTRA
Home
202401090035@mitaoe.ac.in
Support
Logout
1.2.1. Pass or Fail
26:53
Write a Python program that accepts the number of courses and the marks of a student in those courses.
The grade is determined based on the aggregate percentage:
• If the aggregate percentage is greater than 75, the grade is Distinction.
• If the aggregate percentage is greater than or equal to 60 but less than 75, the grade is First Division.
• If the aggregate percentage is greater than or equal to 50 but less than 60, the grade is Second Division.
• If the aggregate percentage is greater than or equal to 40 but less than 50, the grade is Third Division.
Input Format:
The first input will be an integer n, the number of courses.
The second input will be n integers representing the marks of the student in each of the n courses, separated by a space.
Output Format:
If the student passes all courses:
• Print the aggregate percentage (rounded to two decimal places).
• Print the grade based on the aggregate percentage.
If the student fails any course (marks < 40 in any course), print:
Sample Test Cases
passorFa...
Submit
Explorer
Debugger
n = int(input())
marks = list(map(int,input().split(" ")))
if all (mark>=40 for mark in marks):
per = sum(marks)/n
print(f"Aggregate Percentage: {per:.2f}")
if per>=75:
print("Grade: Distinction")
elif 60<=per<75:
print("Grade: First Division")
elif 50<=per<60:
print("Grade: Second Division")
Average time
0.013 s
13.00 ms
Maximum time
0.015 s
15.00 ms
2 out of 2 shown test case(s) passed
2 out of 2 hidden test case(s) passed
Test case 1
11 ms
Debug
Expected output
5
56 78 97 86 93
Aggregate Percentage: 82.00
Grade: Distinction
Actual output
5
56 78 97 86 93
Aggregate Percentage: 82.00
Grade: Distinction
Test case 2
15 ms
Terminal
Test cases
‹ Prev
Reset
Submit
Next ›
Search
ENG
IN
07-05-2025

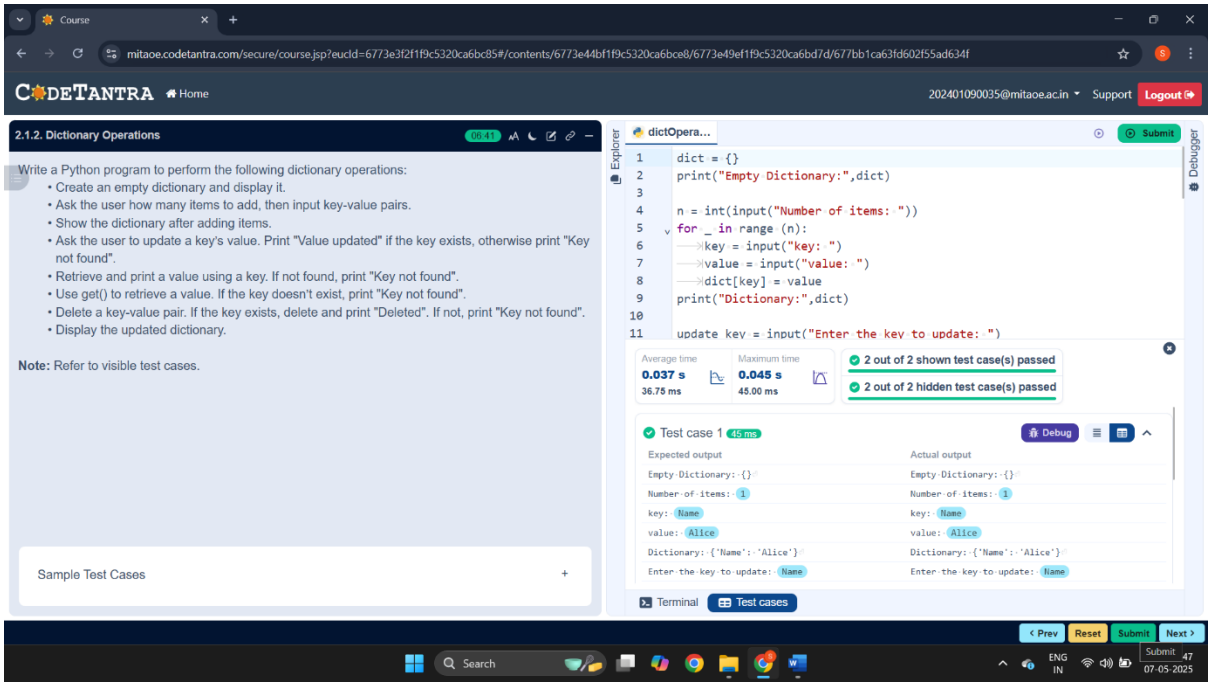

Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e447f1f9c5320ca6bce4/6773e49af1f9c5320ca6bd79/65f7d776bfe2dc3ed510ca60
CODETANTRA
Home
202401090035@mitaoe.ac.in
Support
Logout
1.2.2. Fibonacci series using Recursive Function
03:38
Write a Python program to 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.
Sample Test Cases
fib.py
Submit
Explorer
Debugger
def fib(i):
if(i==0):
return 0
elif(i==1):
return 1
else:
return fib(i-1)+fib(i-2)
Average time
0.005 s
5.00 ms
Maximum time
0.006 s
6.00 ms
2 out of 2 shown test case(s) passed
2 out of 2 hidden test case(s) passed
Test case 1
6 ms
Debug
Expected output
Enter terms for Fibonacci series: 5
0 1 1 2 3
Actual output
Enter terms for Fibonacci series: 5
0 1 1 2 3
Test case 2
4 ms
Terminal
Test cases
‹ Prev
Reset
Submit
Next ›
Search
ENG
IN
07-05-2025

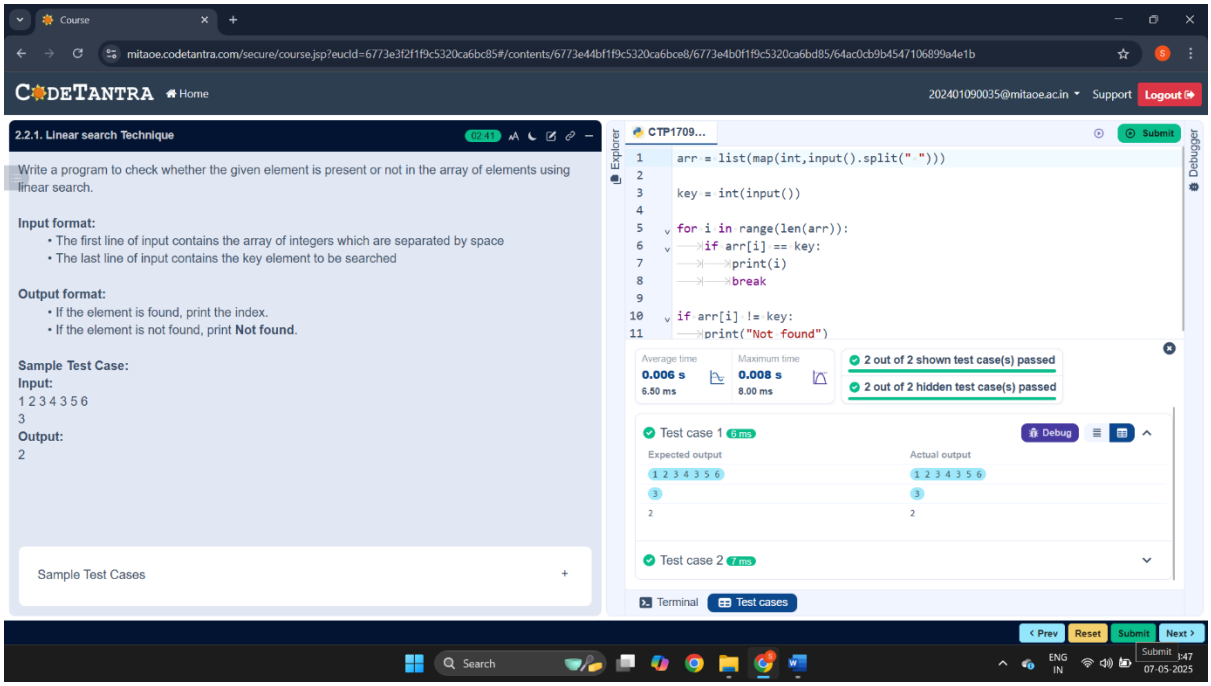

Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e447f1f9c5320ca6bce4/6773e49af1f9c5320ca6bd79/6774c521f4ab787eca9d8a75
CODETANTRA
Home
202401090035@mitaoe.ac.in
Support
Logout
1.2.3. Pattern - 1
01:23
Write a Python program to print a pattern of asterisks in 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...
Submit
Explorer
Debugger
1 a=int(input())
2 for i in range (1,a+1):
3 print('* '*i)
Average time
0.004 s
3.50 ms
Maximum time
0.005 s
5.00 ms
2 out of 2 shown test case(s) passed
4 out of 4 hidden test case(s) passed
Test case 1
5 ms
Debug
Expected output
Actual output
Terminal
Test cases
Prev
Reset
Submit
Next
Search
ENG
IN
07-05-2025

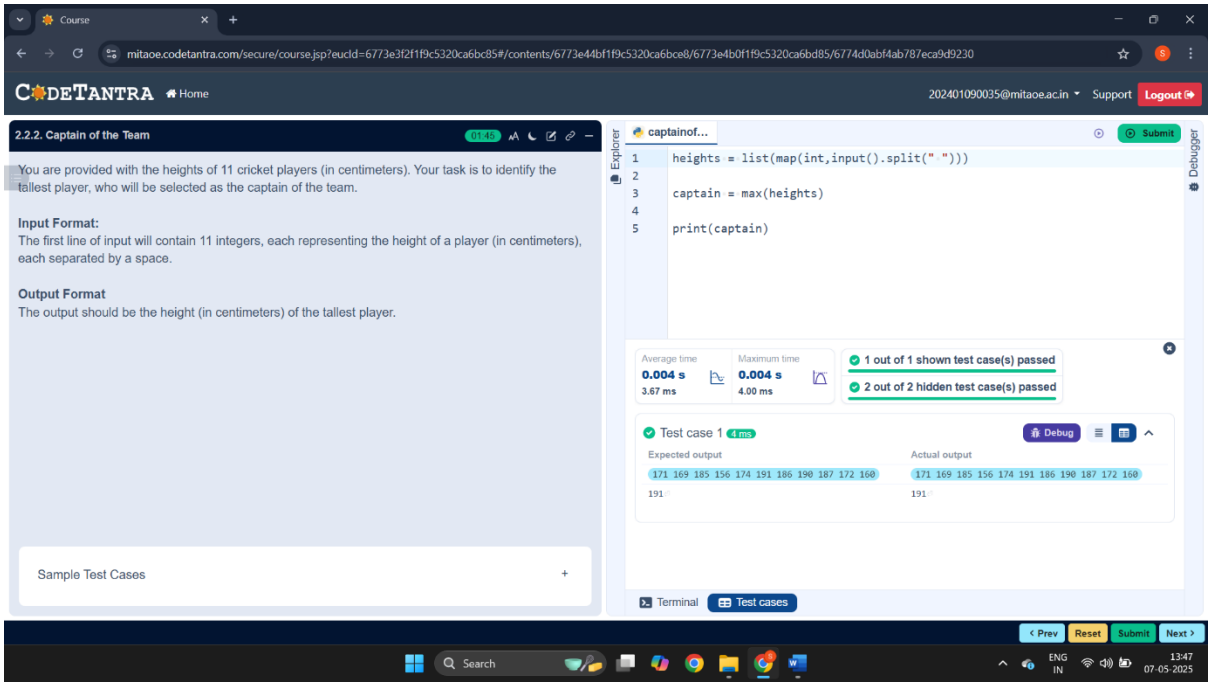

Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e447f1f9c5320ca6bce4/6773e49af1f9c5320ca6bd79/6774cc86f4ab787eca9d8f99
CODETANTRA
Home
202401090035@mitaoe.ac.in
Support
Logout
1.2.4. Pattern - 2
03:19
Write a Python program to print a right-angled triangle 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
numberP...
Submit
Explorer
Debugger
1 n=int(input())
2 i=1
3 while (i<=n):
4 j=1
5 while (j<=i):
6 print(j,end=" ")
7 j+=1
8 print()
9 i+=1
Average time
0.006 s
5.75 ms
Maximum time
0.007 s
7.00 ms
2 out of 2 shown test case(s) passed
2 out of 2 hidden test case(s) passed
Test case 1
6 ms
Debug
Expected output
Actual output
Terminal
Test cases
Prev
Reset
Submit
Next
Search
ENG
IN
13:43
07-05-2025

Practical 2

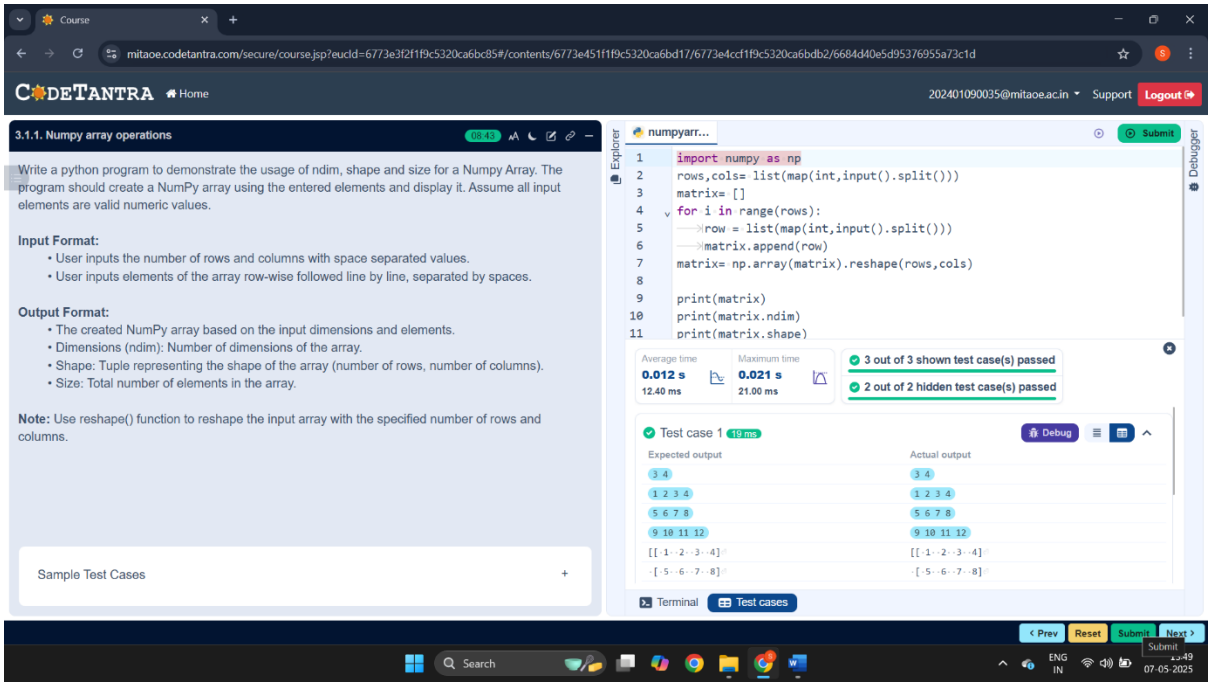

Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e44bf1f9c5320ca6bce8/6773e49ef1f9c5320ca6bd7d/667bbd7f01060f3d690f3f83
CODETANTRA
Home
202401090035@mitaoe.ac.in
Support
Logout
2.1.1. List operations
38:42
Write a Python program that implements a menu-driven interface for managing a list of integers. The program should have the following menu options:
1. Add
2. Remove
3. Display
4. Quit
The program should repeatedly prompt the user to enter a choice from the menu. Depending on the choice selected, the program should perform the following actions:
• Add: Prompts the user to enter an integer and add it to the integer list. If the input is not a valid integer, display "Invalid input".
• Remove: Prompts the user to enter an integer to remove from the list. If the integer is found in the list, remove it; otherwise, display "Element not found". If the list is empty, display "List is empty".
• Display: Displays the current list of integers. If the list is empty, display "List is empty".
• Quit: Exits the program.
• The program should handle invalid menu choices by displaying "Invalid choice".Ensure that the program continues to prompt the user until they choose to quit (option 4).
Sample Test Cases
Explorer
listOps.py
Submit
Debugger
def menu_driven_program():
lst = [] # Initialize an empty list
while True:
print("1. Add")
print("2. Remove")
print("3. Display")
print("4. Quit")
try:
choice = int(input("Enter choice: "))
Average time
0.105 s
105.00 ms
Maximum time
0.168 s
168.00 ms
2 out of 2 shown test case(s) passed
1 out of 1 hidden test case(s) passed
Test case 1
80 ms
Debug
Expected output
Actual output
1. Add
2. Remove
3. Display
4. Quit
Enter choice: 1
Integer: 11
Terminal
Test cases
Prev
Reset
Submit
Next
Search
ENG
IN
13:46
07-05-2025

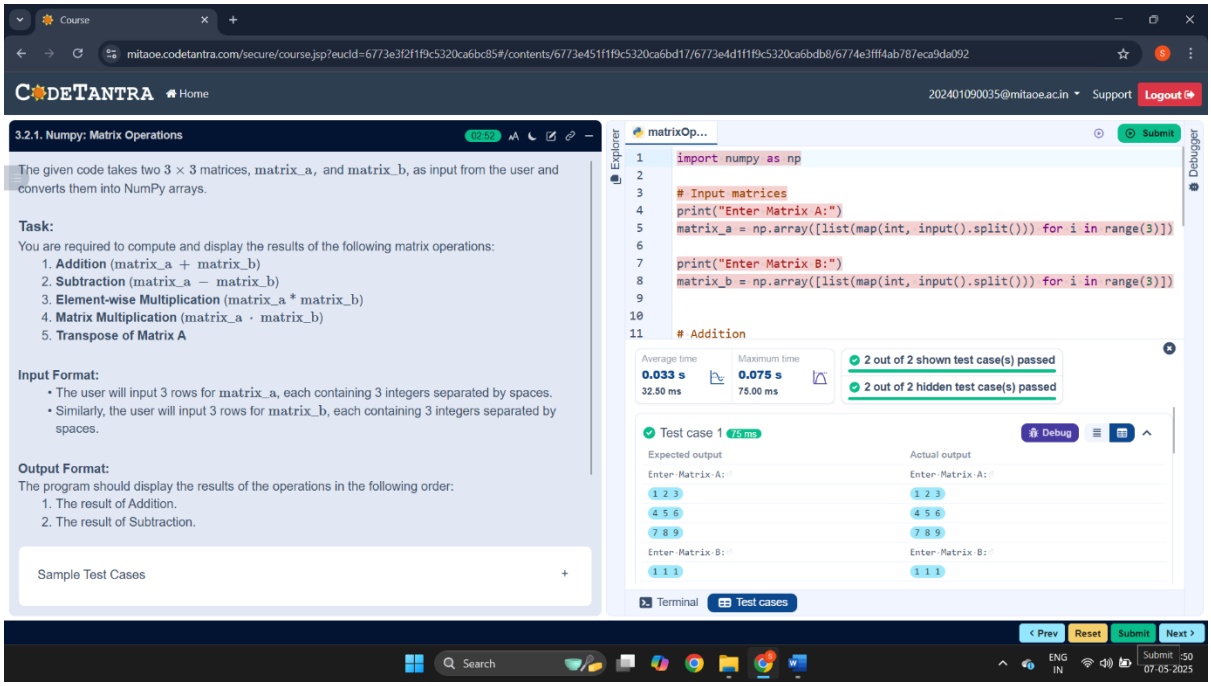

Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e44bf1f9c5320ca6bce8/6773e49ef1f9c5320ca6bd7d/677bb1ca63fd602f55ad634f
CODETANTRA
Home
202401090035@mitaoe.ac.in
Support
Logout
2.1.2. Dictionary Operations
06:41
Write a Python program to perform the following dictionary operations:
• Create an empty dictionary and display it.
• Ask the user how many items to add, then input key-value pairs.
• Show the dictionary after adding items.
• Ask the user to update a key's value. Print "Value updated" if the key exists, otherwise print "Key not found".
• Retrieve and print a value using a key. If not found, print "Key not found".
• Use get() to retrieve a value. If the key doesn't exist, print "Key not found".
• Delete a key-value pair. If the key exists, delete and print "Deleted". If not, print "Key not found".
• Display the updated dictionary.
Note: Refer to visible test cases.
Sample Test Cases
Explorer
dictOpera...
Submit
Debugger
dict = {}
print("Empty Dictionary:",dict)
n = int(input("Number of items: "))
for _ in range (n):
key = input("key: ")
value = input("value: ")
dict[key] = value
print("Dictionary:",dict)
update_key = input("Enter the key to update: ")
Average time
0.037 s
36.75 ms
Maximum time
0.045 s
45.00 ms
2 out of 2 shown test case(s) passed
2 out of 2 hidden test case(s) passed
Test case 1
45 ms
Debug
Expected output
Actual output
Empty Dictionary: {}
Number of items: 1
key: Name
value: Alice
Dictionary: {'Name': 'Alice'}
Enter the key to update: Name
Terminal
Test cases
Prev
Reset
Submit
Next
Search
ENG
IN
Submit
47
07-05-2025

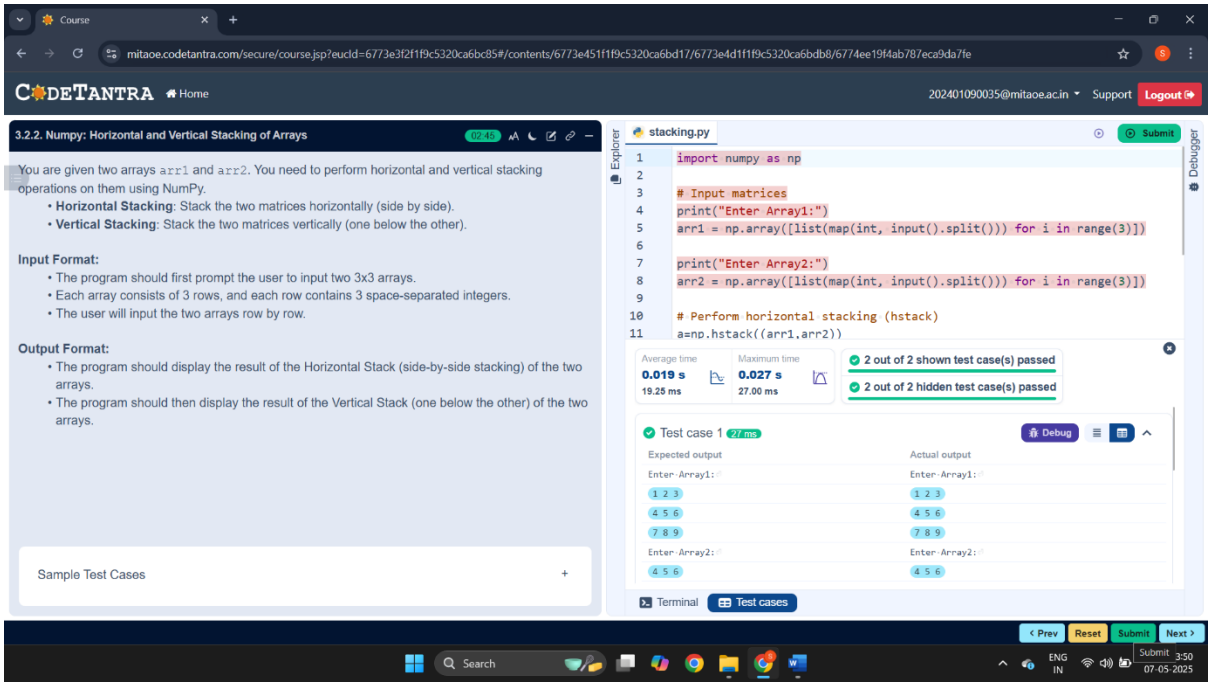

Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e44bf1f9c5320ca6bce8/6773e4b0f1f9c5320ca6bd85/64ac0cb9b4547106899a4e1b
CODETANTRA
Home
202401090035@mitaoe.ac.in
Support
Logout
2.2.1. Linear search Technique
02:41
Write a program to check whether the given element is present or not in the array of elements using linear search.
Input format:
• The first line of input contains the array of integers which are separated by space
• The last line of input contains the key element to be searched
Output format:
• If the element is found, print the index.
• If the element is not found, print Not found.
Sample Test Case:
Input:
1 2 3 4 3 5 6
3
Output:
2
Sample Test Cases
CTP1709...
Submit
Explorer
Debugger
arr = list(map(int,input().split(" ")))
key = int(input())
for i in range(len(arr)):
if arr[i] == key:
print(i)
break
if arr[i] != key:
print("Not found")
Average time
0.006 s
6.50 ms
Maximum time
0.008 s
8.00 ms
2 out of 2 shown test case(s) passed
2 out of 2 hidden test case(s) passed
Test case 1
6 ms
Debug
Expected output
1 2 3 4 3 5 6
3
2
Actual output
1 2 3 4 3 5 6
3
2
Test case 2
7 ms
Terminal
Test cases
‹ Prev
Reset
Submit
Next ›
Search
ENG
IN
13:47
07-05-2025

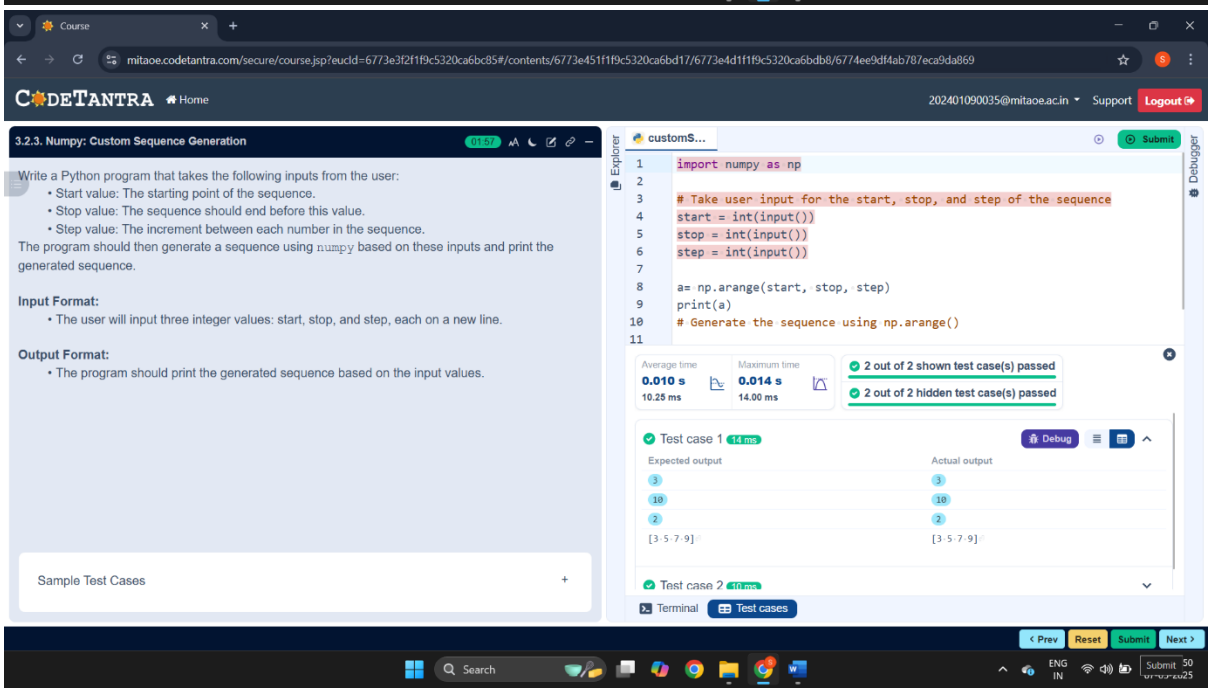

Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e44bf1f9c5320ca6bce8/6773e4b0f1f9c5320ca6bd85/6774d0abf4ab787eca9d9230
CODETANTRA
Home
202401090035@mitaoe.ac.in
Support
Logout
2.2.2. Captain of the Team
01:45
You are provided with the heights of 11 cricket players (in centimeters). Your task is to identify the tallest player, who will be selected as the captain of the team.
Input Format:
The first line of input will contain 11 integers, each representing the height of a player (in centimeters), each separated by a space.
Output Format
The output should be the height (in centimeters) of the tallest player.
Sample Test Cases
captainof...
Submit
Explorer
Debugger
heights = list(map(int,input().split(" ")))
captain = max(heights)
print(captain)
Average time
0.004 s
3.67 ms
Maximum time
0.004 s
4.00 ms
1 out of 1 shown test case(s) passed
2 out of 2 hidden test case(s) passed
Test case 1
4 ms
Debug
Expected output
171 169 185 156 174 191 186 190 187 172 160
191
Actual output
171 169 185 156 174 191 186 190 187 172 160
191
Terminal
Test cases
‹ Prev
Reset
Submit
Next ›
Search
ENG
IN
13:47
07-05-2025

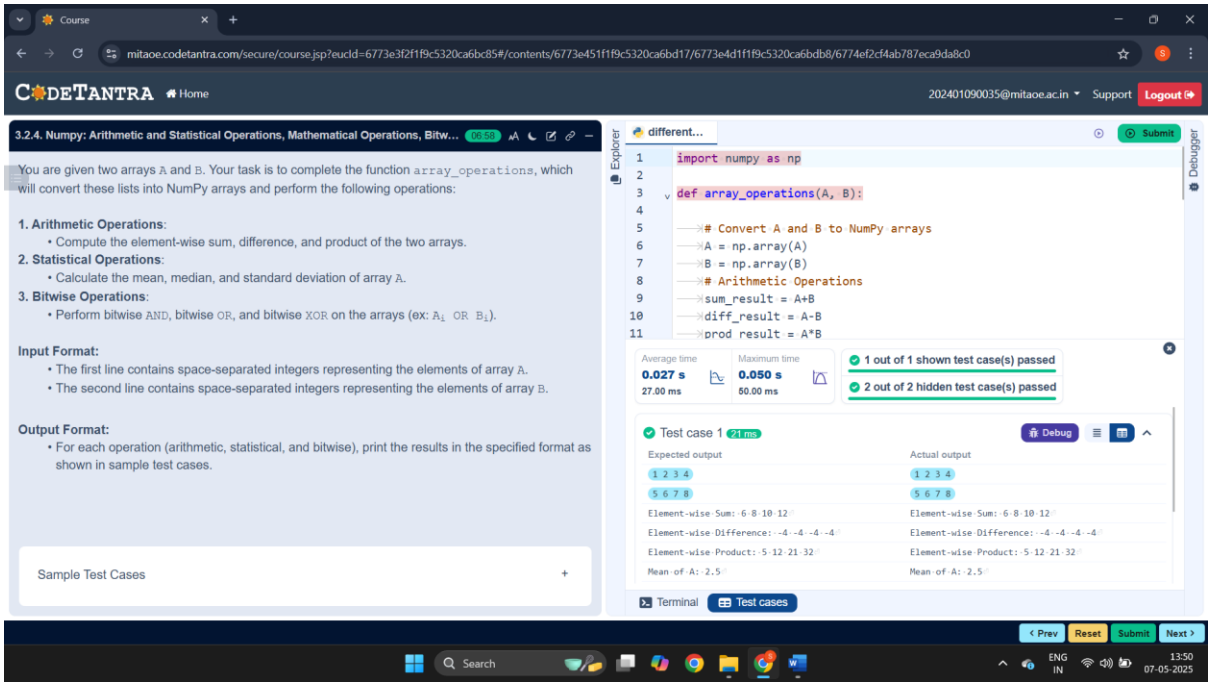
Practical 3

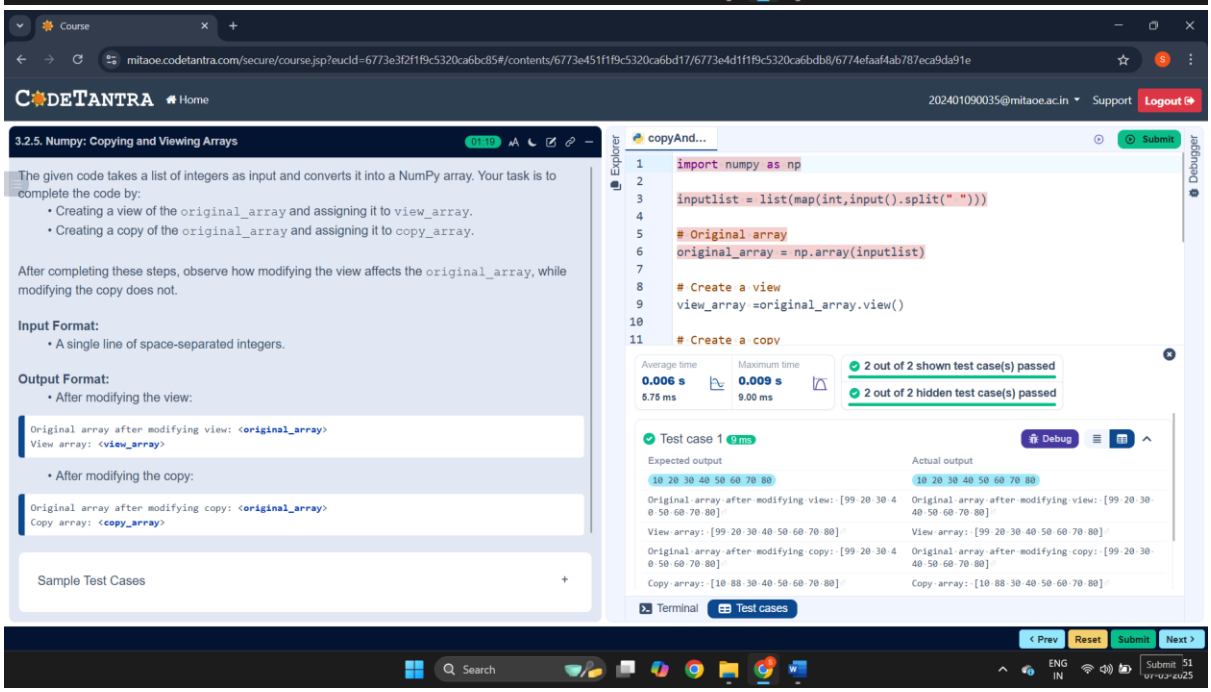

Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e451f1f9c5320ca6bd17/6773e4ccf1f9c5320ca6bdb2/6684d40e5d95376955a73c1d
CODETANTRA
Home
202401090035@mitaoe.ac.in
Support
Logout
3.1.1. Numpy array operations
08:43
Write a python program to demonstrate the usage of ndim, shape and size for a Numpy Array. The program should create a NumPy array using the entered elements and display it. Assume all input elements are valid numeric values.
Input Format:
• User inputs the number of rows and columns with space separated values.
• User inputs elements of the array row-wise followed line by line, separated by spaces.
Output Format:
• The created NumPy array based on the input dimensions and elements.
• Dimensions (ndim): Number of dimensions of the array.
• Shape: Tuple representing the shape of the array (number of rows, number of columns).
• Size: Total number of elements in the array.
Note: Use reshape() function to reshape the input array with the specified number of rows and columns.
Sample Test Cases
Explorer
numpyarr...
Submit
Debugger
import numpy as np
rows,cols= list(map(int,input().split()))
matrix= []
for i in range(rows):
row = list(map(int,input().split()))
matrix.append(row)
matrix= np.array(matrix).reshape(rows,cols)
print(matrix)
print(matrix.ndim)
print(matrix.shape)
Average time
0.012 s
12.40 ms
Maximum time
0.021 s
21.00 ms
3 out of 3 shown test case(s) passed
2 out of 2 hidden test case(s) passed
Test case 1
19 ms
Debug
Expected output
Actual output
3 4
1 2 3 4
5 6 7 8
9 10 11 12
[[1 2 3 4]
[5 6 7 8]]
Terminal
Test cases
Prev
Reset
Submit
Next
Search
ENG
IN
07-05-2025

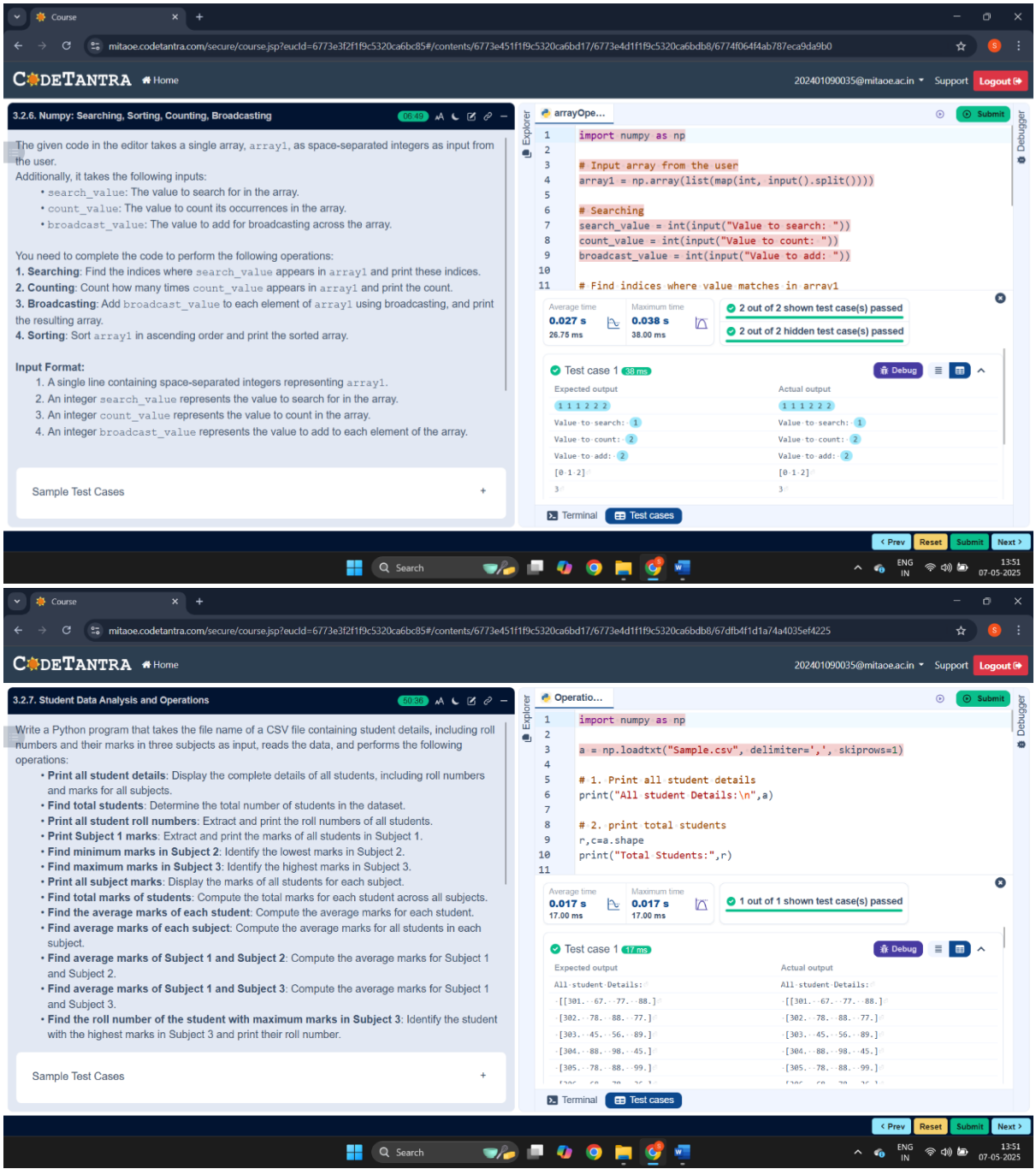

Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e451f1f9c5320ca6bd17/6773e4d1f1f9c5320ca6bdb8/6774e3fff4ab787eca9da092
CODETANTRA
Home
202401090035@mitaoe.ac.in
Support
Logout
3.2.1. Numpy: Matrix Operations
02:52
The given code takes two 3×3 matrices, matrix_a, 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
Explorer
matrixOp...
Submit
Debugger
import numpy as np
# Input matrices
print("Enter Matrix A:")
matrix_a = np.array([list(map(int, input().split())) for i in range(3)])
print("Enter Matrix B:")
matrix_b = np.array([list(map(int, input().split())) for i in range(3)])
# Addition
Average time
0.033 s
32.50 ms
Maximum time
0.075 s
75.00 ms
2 out of 2 shown test case(s) passed
2 out of 2 hidden test case(s) passed
Test case 1
75 ms
Debug
Expected output
Actual output
Enter Matrix A:
1 2 3
4 5 6
7 8 9
Enter Matrix B:
1 1 1
Terminal
Test cases
Prev
Reset
Submit
Next
Search
ENG
IN
07-05-2025


Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e451f1f9c5320ca6bd17/6773e4d1f1f9c5320ca6bdb8/6774ee19f4ab787eca9da7fe
CODETANTRA
Home
202401090035@mitaoe.ac.in
Support
Logout
3.2.2. Numpy: Horizontal and Vertical Stacking of Arrays
02:45
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
Explorer
stacking.py
Submit
Debugger
import numpy as np
# Input matrices
print("Enter Array1:")
arr1 = np.array([list(map(int, input().split())) for i in range(3)])
print("Enter Array2:")
arr2 = np.array([list(map(int, input().split())) for i in range(3)])
# Perform horizontal stacking (hstack)
a=np.hstack((arr1,arr2))
Average time
0.019 s
19.25 ms
Maximum time
0.027 s
27.00 ms
2 out of 2 shown test case(s) passed
2 out of 2 hidden test case(s) passed
Test case 1
27 ms
Debug
Expected output
Actual output
Enter Array1:
1 2 3
4 5 6
7 8 9
Enter Array2:
4 5 6
Terminal
Test cases
‹ Prev
Reset
Submit
Next ›
Search
ENG
IN
3:50
07-05-2025

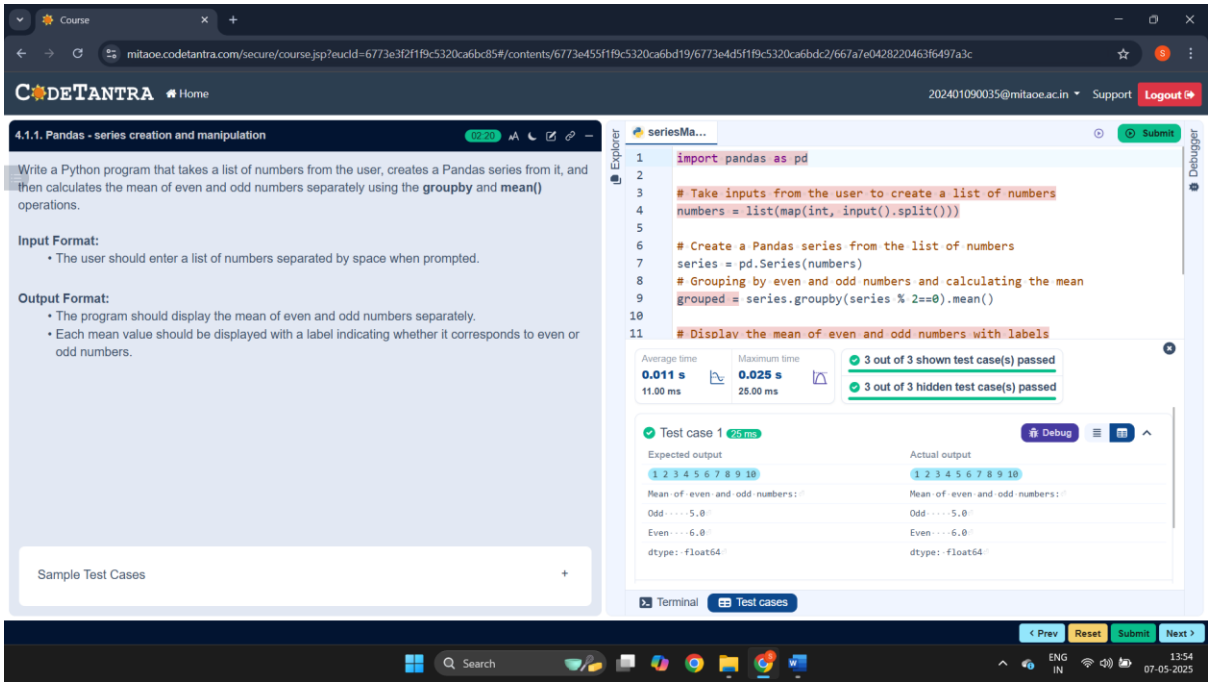

Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e451f1f9c5320ca6bd17/6773e4d1f1f9c5320ca6bdb8/6774ee9df4ab787eca9da869
CODETANTRA
Home
202401090035@mitaoe.ac.in
Support
Logout
3.2.3. Numpy: Custom Sequence Generation
01:57
Write a Python program that takes 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.
Sample Test Cases
Explorer
customS...
Submit
Debugger
import numpy as np
# Take user input for the start, stop, and step of the sequence
start = int(input())
stop = int(input())
step = int(input())
a= np.arange(start, stop, step)
print(a)
# Generate the sequence using np.arange()
Average time
0.010 s
10.25 ms
Maximum time
0.014 s
14.00 ms
2 out of 2 shown test case(s) passed
2 out of 2 hidden test case(s) passed
Test case 1
14 ms
Debug
Expected output
Actual output
3
10
2
[3 5 7 9]
Test case 2
Terminal
Test cases
‹ Prev
Reset
Submit
Next ›
Search
ENG
IN
Submit

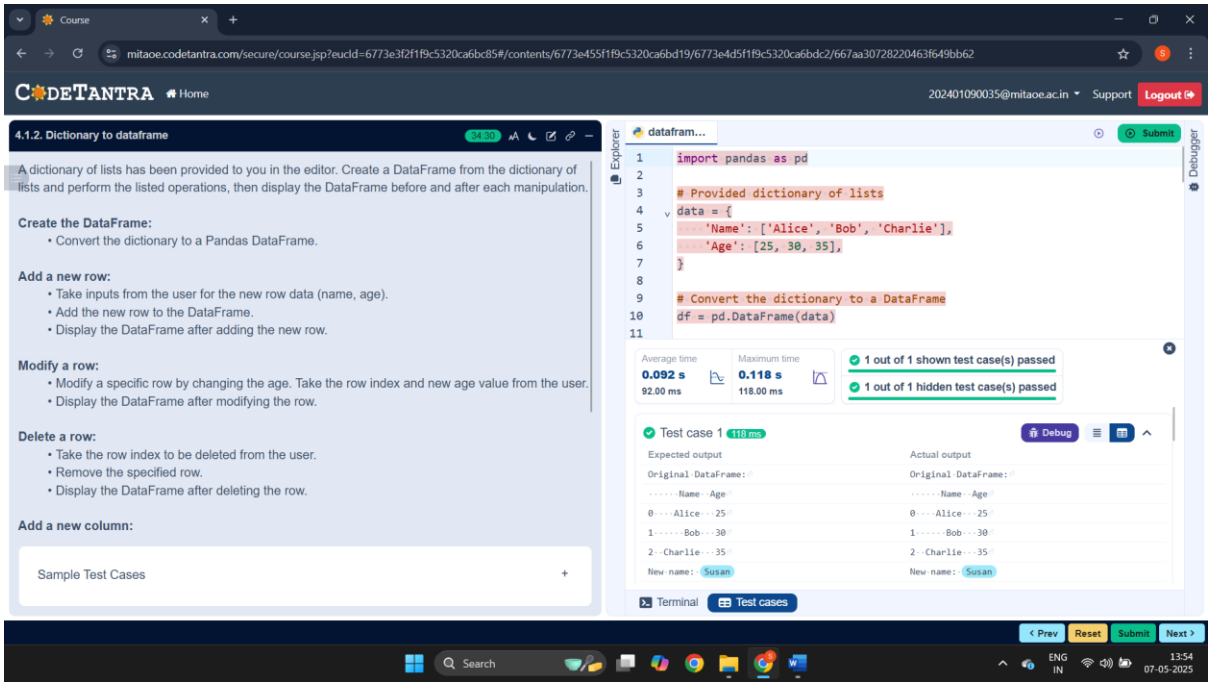

Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e451f1f9c5320ca6bd17/6773e4d1f1f9c5320ca6bdb8/6774ef2cf4ab787eca9da8c0
CODETANTRA
Home
202401090035@mitaoe.ac.in
Support
Logout
3.2.4. Numpy: Arithmetic and Statistical Operations, Mathematical Operations, Bitw...
06:58
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 A.
• The second line contains space-separated integers representing the elements of array B.
Output Format:
• For each operation (arithmetic, statistical, and bitwise), print the results in the specified format as shown in sample test cases.
Sample Test Cases
Explorer
different...
Submit
Debugger
import numpy as np
def array_operations(A, B):
# Convert A and B to NumPy arrays
A = np.array(A)
B = np.array(B)
# Arithmetic Operations
sum_result = A+B
diff_result = A-B
prod_result = A*B
Average time
0.027 s
27.00 ms
Maximum time
0.050 s
50.00 ms
1 out of 1 shown test case(s) passed
2 out of 2 hidden test case(s) passed
Test case 1
21 ms
Debug
Expected output
Actual output
1 2 3 4
5 6 7 8
Element-wise Sum: 6 8 10 12
Element-wise Difference: -4 -4 -4 -4
Element-wise Product: 5 12 21 32
Mean of A: 2.5
Terminal
Test cases
Prev
Reset
Submit
Next
Search
ENG
IN
13:50
07-05-2025

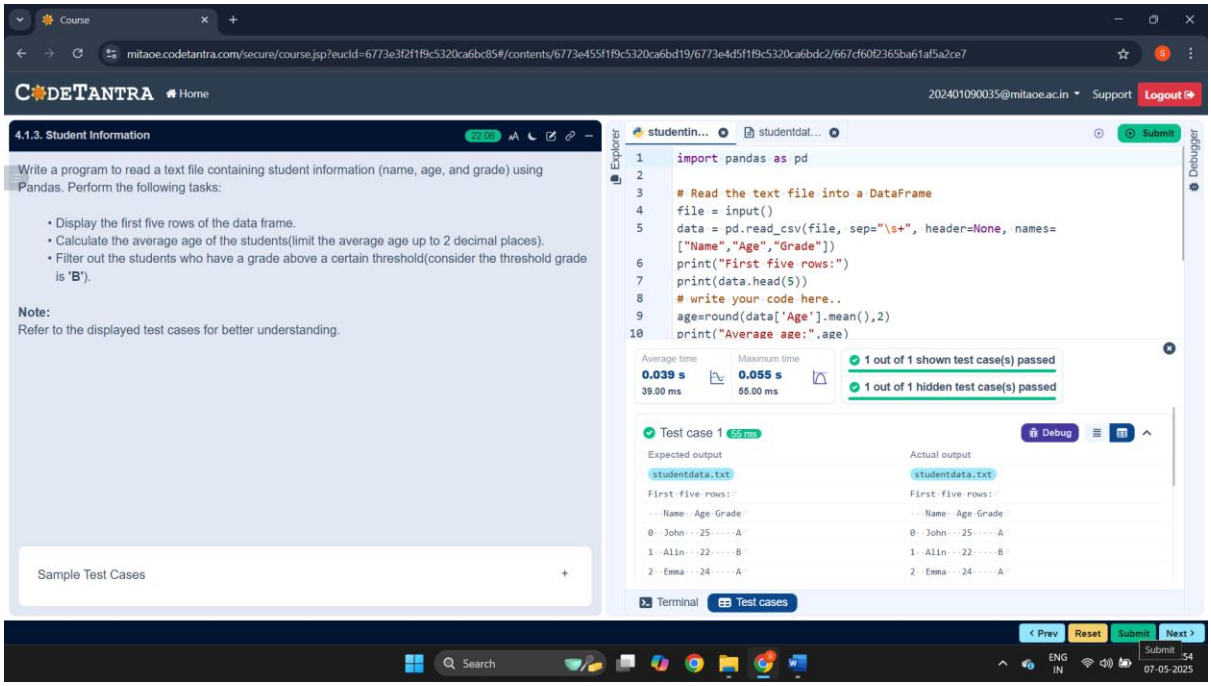

Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e451f1f9c5320ca6bd17/6773e4d1f1f9c5320ca6bdb8/6774efaaf4ab787eca9da91e
CODETANTRA
Home
202401090035@mitaoe.ac.in
Support
Logout
3.2.5. Numpy: Copying and Viewing Arrays
01:19
The given code takes a list of integers 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
Explorer
copyAnd...
Submit
Debugger
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
Average time
0.006 s
6.75 ms
Maximum time
0.009 s
9.00 ms
2 out of 2 shown test case(s) passed
2 out of 2 hidden test case(s) passed
Test case 1
9 ms
Debug
Expected output
Actual output
10 20 30 40 50 60 70 80
Original array after modifying view: [99 20 30 40 50 60 70 80]
View array: [99 20 30 40 50 60 70 80]
Original array after modifying copy: [99 20 30 40 50 60 70 80]
Copy array: [10 88 30 40 50 60 70 80]
Terminal
Test cases
Prev
Reset
Submit
Next
Search
ENG
IN
Submit 51

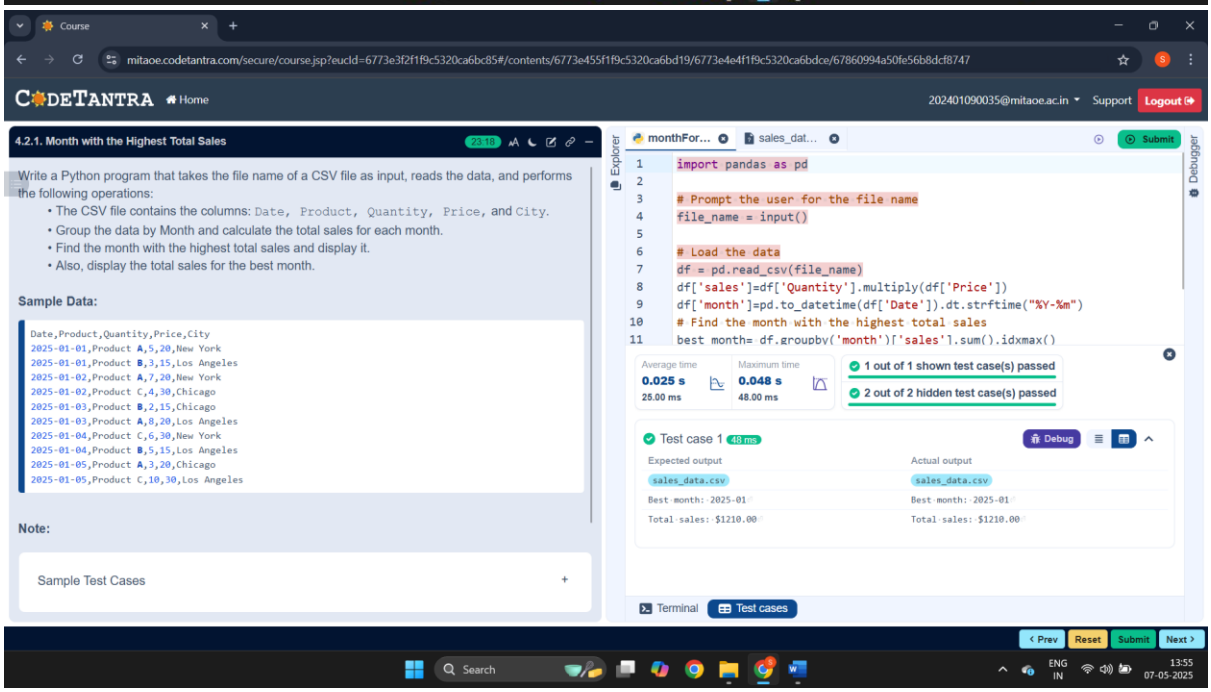

Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e451f1f9c5320ca6bd17/6773e4d1f1f9c5320ca6bdb8/6774f064f4ab787eca9da9b0
CODETANTRA
Home
202401090035@mitaoe.ac.in
Support
Logout
3.2.6. Numpy: Searching, Sorting, Counting, Broadcasting
06:49
The given code in 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...
Submit
Explorer
Debugger
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
Average time
0.027 s
26.75 ms
Maximum time
0.038 s
38.00 ms
2 out of 2 shown test case(s) passed
2 out of 2 hidden test case(s) passed
Test case 1
38 ms
Debug
Expected output
Actual output
1 1 1 2 2 2
Value to search: 1
Value to count: 2
Value to add: 2
[0 1 2]
3
Terminal
Test cases
Prev
Reset
Submit
Next
Search
ENG
IN
13:51
07-05-2025
Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e451f1f9c5320ca6bd17/6773e4d1f1f9c5320ca6bdb8/67dfb4f1d1a74a4035ef4225
CODETANTRA
Home
202401090035@mitaoe.ac.in
Support
Logout
3.2.7. Student Data Analysis and Operations
50:36
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.
• Find average marks of Subject 1 and Subject 2: Compute the average marks for Subject 1 and Subject 2.
• Find average marks of Subject 1 and Subject 3: Compute the average marks for Subject 1 and Subject 3.
• Find the roll number of the student with maximum marks in Subject 3: Identify the student with the highest marks in Subject 3 and print their roll number.
Sample Test Cases
Operatio...
Submit
Explorer
Debugger
import numpy as np
a = np.loadtxt("Sample.csv", delimiter=',', skiprows=1)
# 1. Print all student details
print("All student Details:\n",a)
# 2. print total students
r,c=a.shape
print("Total Students:",r)
Average time
0.017 s
17.00 ms
Maximum time
0.017 s
17.00 ms
1 out of 1 shown test case(s) passed
Test case 1
17 ms
Debug
Expected output
Actual output
All student Details:
[[301. 67. 77. 88.]
[302. 78. 88. 77.]
[303. 45. 56. 89.]
[304. 88. 98. 45.]
[305. 78. 88. 99.]
Terminal
Test cases
Prev
Reset
Submit
Next
Search
ENG
IN
13:51
07-05-2025

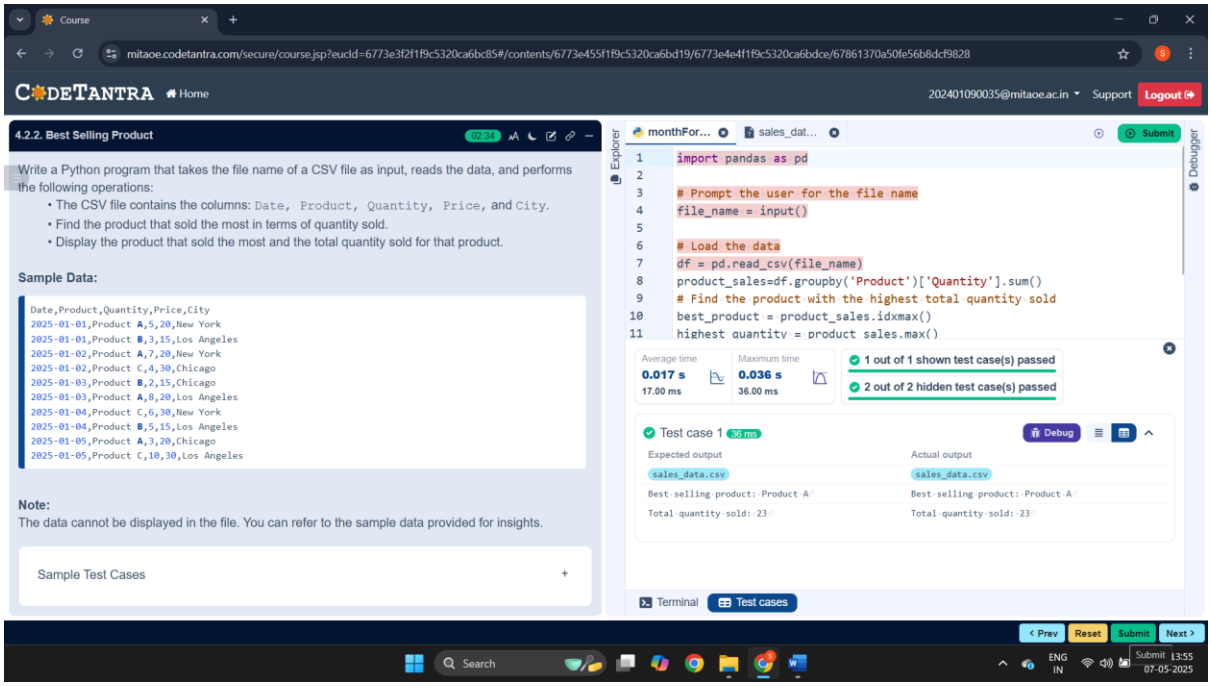
Practical 4

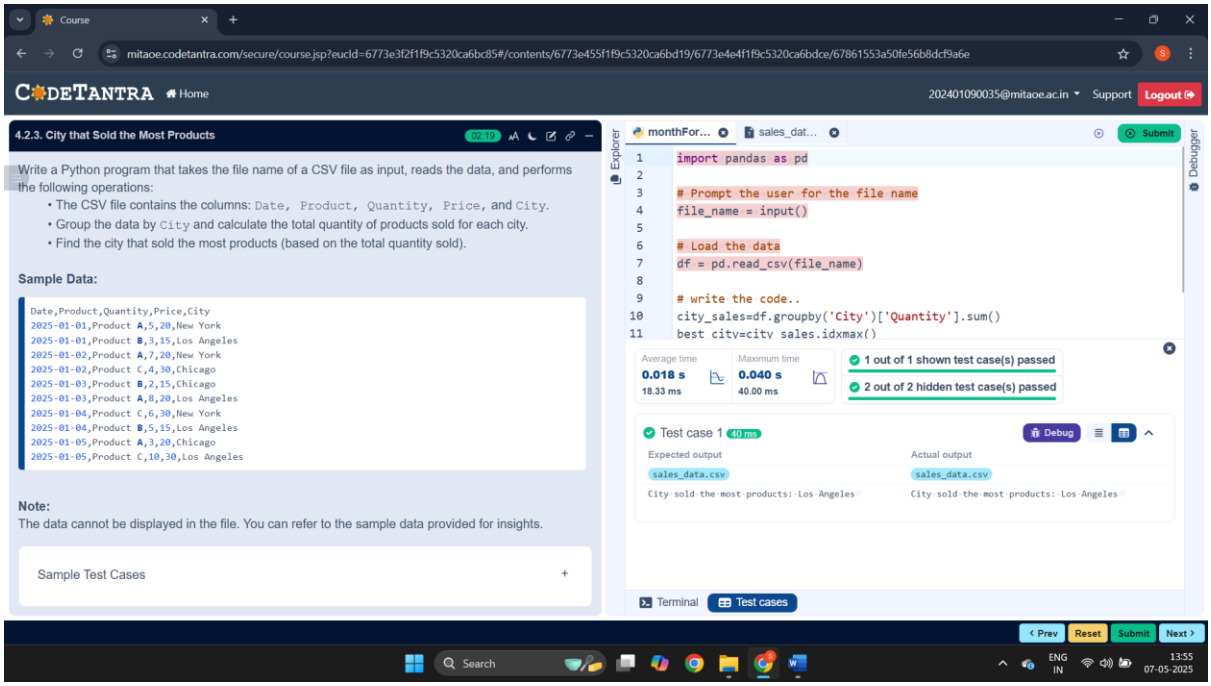

Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e455f1f9c5320ca6bd19/6773e4d5f1f9c5320ca6bdc2/667a7e0428220463f6497a3c
CODETANTRA
Home
202401090035@mitaoe.ac.in
Support
Logout
4.1.1. Pandas - series creation and manipulation
02:20
Write a Python program that takes a list of numbers from the user, creates a Pandas series from it, and then calculates the mean of even and odd numbers separately using the groupby and mean() operations.
Input Format:
• The user should enter a list of numbers separated by space when prompted.
Output Format:
• The program should display the mean of even and odd numbers separately.
• Each mean value should be displayed with a label indicating whether it corresponds to even or odd numbers.
Sample Test Cases
seriesMa...
Submit
Explorer
Debugger
import pandas as pd
# Take inputs from the user to create a list of numbers
numbers = list(map(int, input().split()))
# Create a Pandas series from the list of numbers
series = pd.Series(numbers)
# Grouping by even and odd numbers and calculating the mean
grouped = series.groupby(series % 2==0).mean()
# Display the mean of even and odd numbers with labels
Average time
0.011 s
11.00 ms
Maximum time
0.025 s
25.00 ms
3 out of 3 shown test case(s) passed
3 out of 3 hidden test case(s) passed
Test case 1
25 ms
Debug
Expected output
1 2 3 4 5 6 7 8 9 10
Mean of even and odd numbers:
Odd 5.0
Even 6.0
dtype: float64
Actual output
1 2 3 4 5 6 7 8 9 10
Mean of even and odd numbers:
Odd 5.0
Even 6.0
dtype: float64
Terminal
Test cases
Prev
Reset
Submit
Next
Search
ENG
IN
13:54
07-05-2025

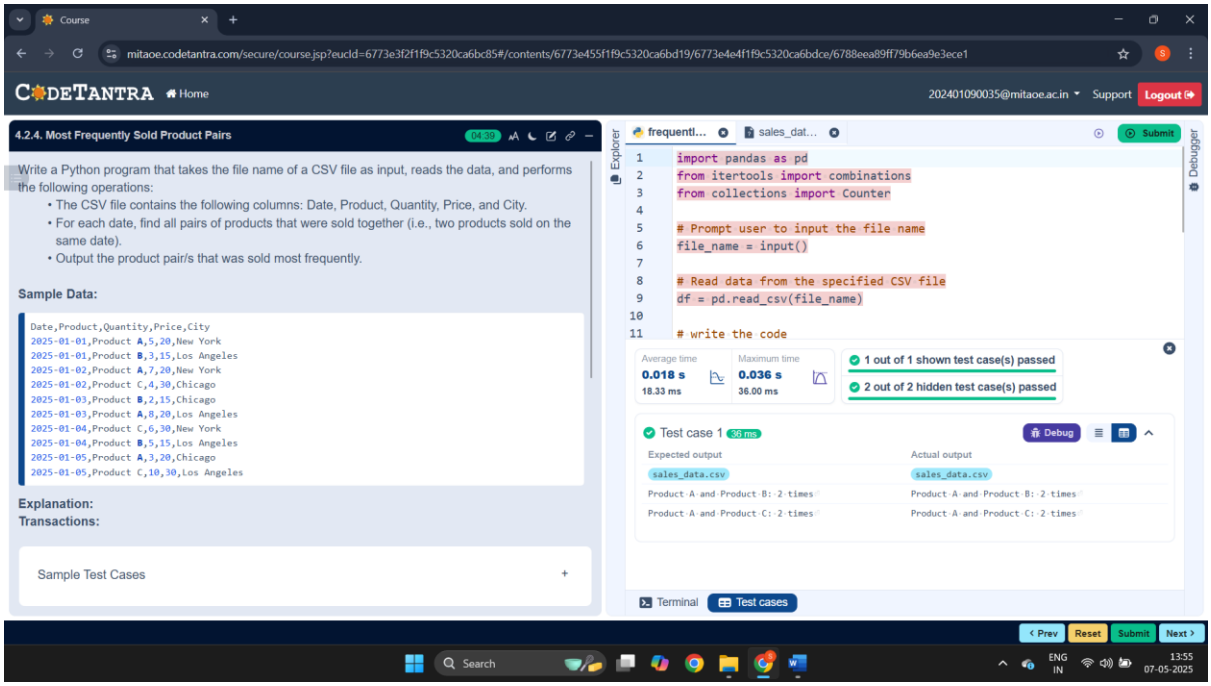

Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e455f1f9c5320ca6bd19/6773e4d5f1f9c5320ca6bdc2/667aa30728220463f649bb62
CODETANTRA
Home
202401090035@mitaoe.ac.in
Support
Logout
4.1.2. Dictionary to dataframe
34:30
A dictionary of lists has been provided to you in the editor. Create a DataFrame from the dictionary of lists and perform the listed operations, then display the DataFrame before and after each manipulation.
Create the DataFrame:
• Convert the dictionary to a Pandas DataFrame.
Add a new row:
• Take inputs from the user for the new row data (name, age).
• Add the new row to the DataFrame.
• Display the DataFrame after adding the new row.
Modify a row:
• Modify a specific row by changing the age. Take the row index and new age value from the user.
• Display the DataFrame after modifying the row.
Delete a row:
• Take the row index to be deleted from the user.
• Remove the specified row.
• Display the DataFrame after deleting the row.
Add a new column:
Sample Test Cases
datafram...
Submit
Explorer
Debugger
import pandas as pd
# Provided dictionary of lists
data = {
'Name': ['Alice', 'Bob', 'Charlie'],
'Age': [25, 30, 35],
}
# Convert the dictionary to a DataFrame
df = pd.DataFrame(data)
Average time
0.092 s
92.00 ms
Maximum time
0.118 s
118.00 ms
1 out of 1 shown test case(s) passed
1 out of 1 hidden test case(s) passed
Test case 1
118 ms
Debug
Expected output
Original DataFrame:
Name Age
0 Alice 25
1 Bob 30
2 Charlie 35
New name: Susan
Actual output
Original DataFrame:
Name Age
0 Alice 25
1 Bob 30
2 Charlie 35
New name: Susan
Terminal
Test cases
Prev
Reset
Submit
Next
Search
ENG
IN
13:54
07-05-2025

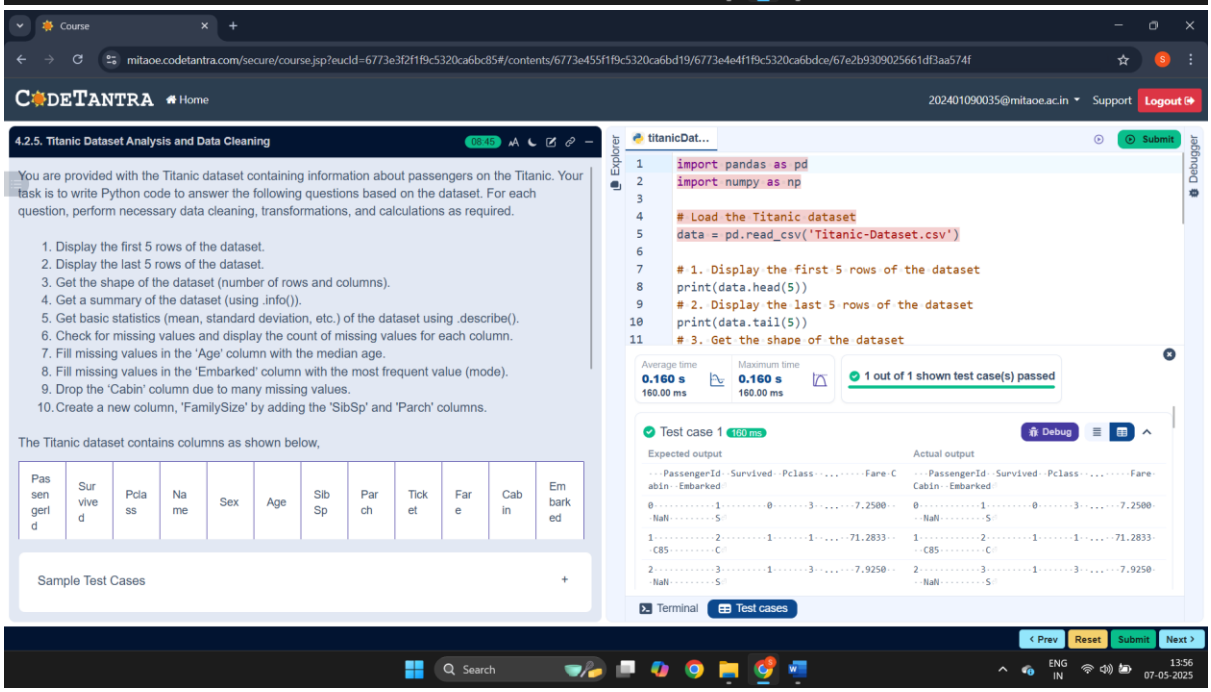

Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e455f1f9c5320ca6bd19/6773e4d5f1f9c5320ca6bdc2/667cf60f2365ba61af5a2ce7
CODETANTRA
Home
202401090035@mitaoe.ac.in
Support
Logout
4.1.3. Student Information
22:06
Write a program to read a text file containing student information (name, age, and grade) using Pandas. Perform the following tasks:
• Display the first five rows of the data frame.
• Calculate the average age of the students(limit the average age up to 2 decimal places).
• Filter out the students who have a grade above a certain threshold(consider the threshold grade is 'B').
Note:
Refer to the displayed test cases for better understanding.
Sample Test Cases
studentin...
studentdat...
Submit
Explorer
Debugger
import pandas as pd
# Read the text file into a DataFrame
file = input()
data = pd.read_csv(file, sep="\s+", header=None, names=["Name","Age","Grade"])
print("First five rows:")
print(data.head(5))
# write your code here..
age=round(data['Age'].mean(),2)
print("Average age:",age)
Average time
0.039 s
39.00 ms
Maximum time
0.055 s
55.00 ms
1 out of 1 shown test case(s) passed
1 out of 1 hidden test case(s) passed
Test case 1
55 ms
Debug
Expected output
studentdata.txt
First five rows:
Name Age Grade
0 John 25 A
1 Alin 22 B
2 Emma 24 A
Actual output
studentdata.txt
First five rows:
Name Age Grade
0 John 25 A
1 Alin 22 B
2 Emma 24 A
Terminal
Test cases
Prev
Reset
Submit
Next
Search
ENG
IN
07-05-2025

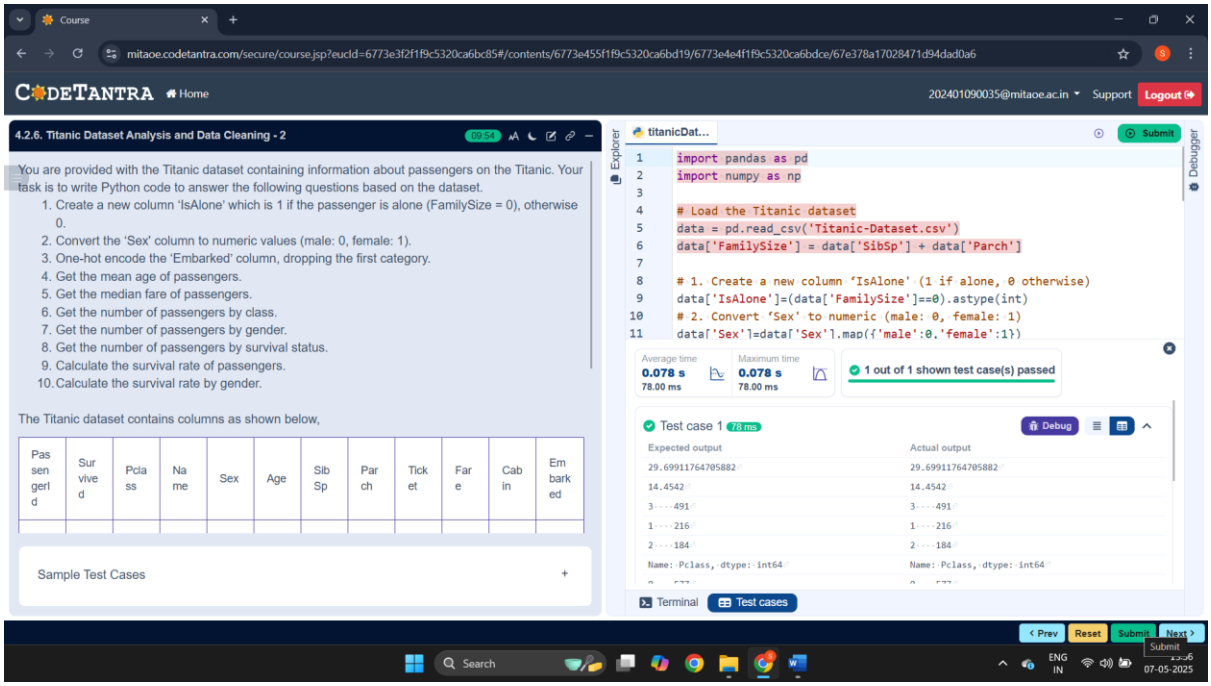

Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e4e4f1f9c5320ca6bdce/67860994a50fe56b8dcf8747
CODETANTRA
Home
202401090035@mitaoe.ac.in
Support
Logout
4.2.1. Month with the Highest Total Sales
23:18
Write a Python program that takes the file name of a CSV file as input, reads the data, and performs the following operations:
• The CSV file contains the columns: Date, Product, Quantity, Price, and City.
• Group the data by Month and calculate the total sales for each month.
• Find the month with the highest total sales and display it.
• Also, display the total sales for the best month.
Sample Data:
Date,Product,Quantity,Price,City
2025-01-01,Product A,5,20,New York
2025-01-01,Product B,3,15,Los Angeles
2025-01-02,Product A,7,20,New York
2025-01-02,Product C,4,30,Chicago
2025-01-03,Product B,2,15,Chicago
2025-01-03,Product A,8,20,Los Angeles
2025-01-04,Product C,6,30,New York
2025-01-04,Product B,5,15,Los Angeles
2025-01-05,Product A,3,20,Chicago
2025-01-05,Product C,10,30,Los Angeles
Note:
Sample Test Cases
monthFor...
sales_dat...
Submit
Explorer
Debugger
import pandas as pd
# Prompt the user for the file name
file_name = input()
# Load the data
df = pd.read_csv(file_name)
df['sales']=df['Quantity'].multiply(df['Price'])
df['month']=pd.to_datetime(df['Date']).dt.strftime("%Y-%m")
# Find the month with the highest total sales
best_month= df.groupby('month')['sales'].sum().idxmax()
Average time
0.025 s
25.00 ms
Maximum time
0.048 s
48.00 ms
1 out of 1 shown test case(s) passed
2 out of 2 hidden test case(s) passed
Test case 1
48 ms
Debug
Expected output
sales_data.csv
Best month: 2025-01
Total sales: $1210.00
Actual output
sales_data.csv
Best month: 2025-01
Total sales: $1210.00
Terminal
Test cases
Prev
Reset
Submit
Next
Search
ENG
IN
13:55
07-05-2025

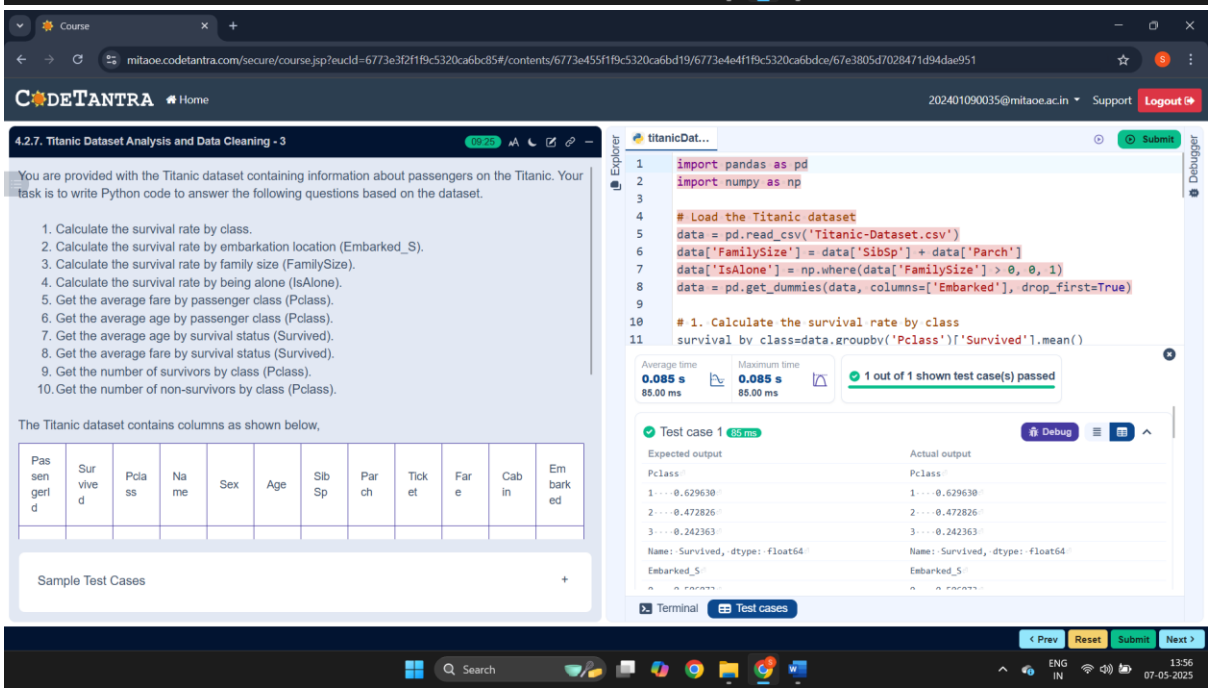

Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e455f1f9c5320ca6bd19/6773e4e4f1f9c5320ca6bdce/67861370a50fe56b8dcf9828
CODETANTRA
Home
202401090035@mitaoe.ac.in
Support
Logout
4.2.2. Best Selling Product
02:34
Write a Python program that takes the file name of a CSV file as input, reads the data, and performs the following operations:
• The CSV file contains the columns: Date, Product, Quantity, Price, and City.
• Find the product that sold the most in terms of quantity sold.
• Display the product that sold the most and the total quantity sold for that product.
Sample Data:
Date,Product,Quantity,Price,City
2025-01-01,Product A,5,20,New York
2025-01-01,Product B,3,15,Los Angeles
2025-01-02,Product A,7,20,New York
2025-01-02,Product C,4,30,Chicago
2025-01-03,Product B,2,15,Chicago
2025-01-03,Product A,8,20,Los Angeles
2025-01-04,Product C,6,30,New York
2025-01-04,Product B,5,15,Los Angeles
2025-01-05,Product A,3,20,Chicago
2025-01-05,Product C,10,30,Los Angeles
Note:
The data cannot be displayed in the file. You can refer to the sample data provided for insights.
Sample Test Cases
Explorer
monthFor...
sales_dat...
Submit
Debugger
import pandas as pd
# Prompt the user for the file name
file_name = input()
# Load the data
df = pd.read_csv(file_name)
product_sales=df.groupby('Product')['Quantity'].sum()
# Find the product with the highest total quantity sold
best_product = product_sales.idxmax()
highest_quantity = product_sales.max()
Average time
0.017 s
17.00 ms
Maximum time
0.036 s
36.00 ms
1 out of 1 shown test case(s) passed
2 out of 2 hidden test case(s) passed
Test case 1
36 ms
Debug
Expected output
sales_data.csv
Best selling product: Product A
Total quantity sold: 23
Actual output
sales_data.csv
Best selling product: Product A
Total quantity sold: 23
Terminal
Test cases
Prev
Reset
Submit
Next
Search
ENG
IN
13:55
07-05-2025

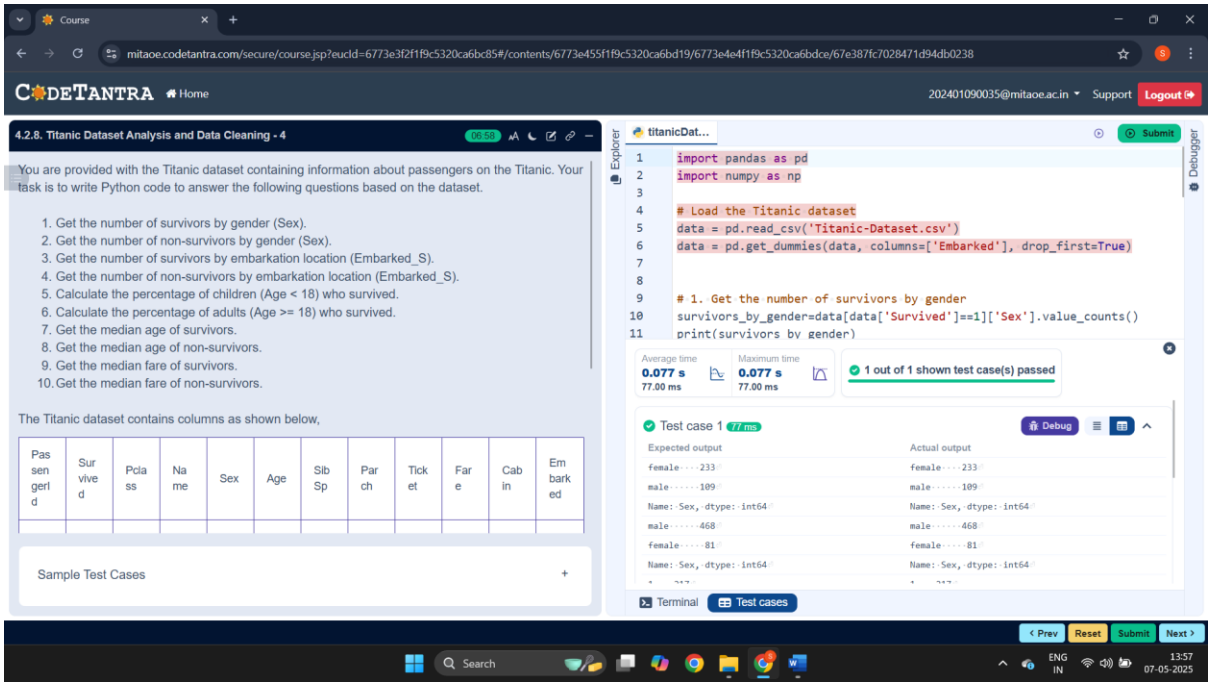

Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e455f1f9c5320ca6bd19/6773e4e4f1f9c5320ca6bdce/67861553a50fe56b8dcf9a6e
CODETANTRA
Home
202401090035@mitaoe.ac.in
Support
Logout
4.2.3. City that Sold the Most Products
02:19
Write a Python program that takes the file name of a CSV file as input, reads the data, and performs the following operations:
• The CSV file contains the columns: Date, Product, Quantity, Price, and City.
• Group the data by City and calculate the total quantity of products sold for each city.
• Find the city that sold the most products (based on the total quantity sold).
Sample Data:
Date,Product,Quantity,Price,City
2025-01-01,Product A,5,20,New York
2025-01-01,Product B,3,15,Los Angeles
2025-01-02,Product A,7,20,New York
2025-01-02,Product C,4,30,Chicago
2025-01-03,Product B,2,15,Chicago
2025-01-03,Product A,8,20,Los Angeles
2025-01-04,Product C,6,30,New York
2025-01-04,Product B,5,15,Los Angeles
2025-01-05,Product A,3,20,Chicago
2025-01-05,Product C,10,30,Los Angeles
Note:
The data cannot be displayed in the file. You can refer to the sample data provided for insights.
Sample Test Cases
Explorer
monthFor...
sales_dat...
Submit
Debugger
import pandas as pd
# Prompt the user for the file name
file_name = input()
# Load the data
df = pd.read_csv(file_name)
# write the code..
city_sales=df.groupby('City')['Quantity'].sum()
best_city=city_sales.idxmax()
Average time
0.018 s
18.33 ms
Maximum time
0.040 s
40.00 ms
1 out of 1 shown test case(s) passed
2 out of 2 hidden test case(s) passed
Test case 1
40 ms
Debug
Expected output
sales_data.csv
City sold the most products: Los Angeles
Actual output
sales_data.csv
City sold the most products: Los Angeles
Terminal
Test cases
Prev
Reset
Submit
Next
Search
ENG
IN
13:55
07-05-2025


Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e455f1f9c5320ca6bd19/6773e4e4f1f9c5320ca6bdce/6788eea89ff79b6ea9e3ece1
CODETANTRA
Home
202401090035@mitaoe.ac.in
Support
Logout
4.2.4. Most Frequently Sold Product Pairs
04:39
Write a Python program that takes the file name of a CSV file as input, reads the data, and performs the following operations:
• The CSV file contains the following columns: Date, Product, Quantity, Price, and City.
• For each date, find all pairs of products that were sold together (i.e., two products sold on the same date).
• Output the product pair/s that was sold most frequently.
Sample Data:
Date,Product,Quantity,Price,City
2025-01-01,Product A,5,20,New York
2025-01-01,Product B,3,15,Los Angeles
2025-01-02,Product A,7,20,New York
2025-01-02,Product C,4,30,Chicago
2025-01-03,Product B,2,15,Chicago
2025-01-03,Product A,8,20,Los Angeles
2025-01-04,Product C,6,30,New York
2025-01-04,Product B,5,15,Los Angeles
2025-01-05,Product A,3,20,Chicago
2025-01-05,Product C,10,30,Los Angeles
Explanation:
Transactions:
Sample Test Cases
Explorer
frequentl...
sales_dat...
Submit
Debugger
import pandas as pd
from itertools import combinations
from collections import Counter
# Prompt user to input the file name
file_name = input()
# Read data from the specified CSV file
df = pd.read_csv(file_name)
# write the code
Average time
0.018 s
18.33 ms
Maximum time
0.036 s
36.00 ms
1 out of 1 shown test case(s) passed
2 out of 2 hidden test case(s) passed
Test case 1
36 ms
Debug
Expected output
sales_data.csv
Product A and Product B: 2 times
Product A and Product C: 2 times
Actual output
sales_data.csv
Product A and Product B: 2 times
Product A and Product C: 2 times
Terminal
Test cases
Prev
Reset
Submit
Next
Search
ENG
IN
13:55
07-05-2025

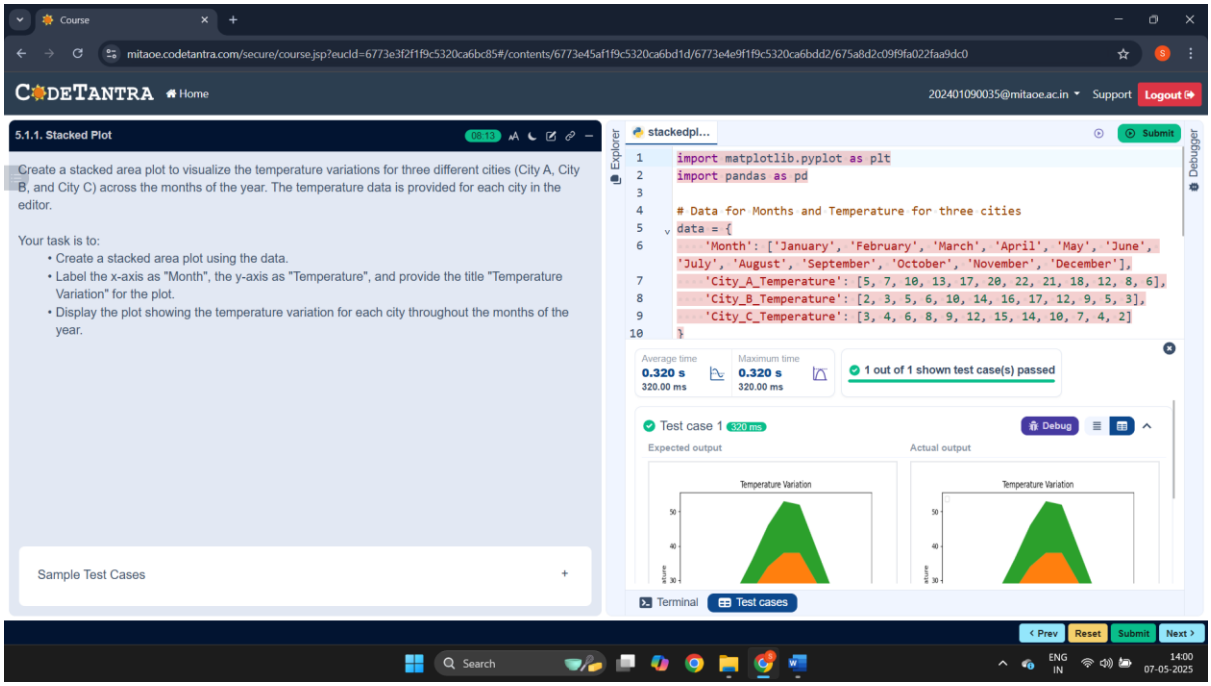

Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e455f1f9c5320ca6bd19/6773e4e4f1f9c5320ca6bdce/67e2b9309025661df3aa574f
CODETANTRA
Home
202401090035@mitaoe.ac.in
Support
Logout
4.2.5. Titanic Dataset Analysis and Data Cleaning
08:45
You are provided with the Titanic dataset containing information about passengers on the Titanic. Your task is to write Python code to answer the following questions based on the dataset. For each question, perform necessary data cleaning, transformations, and calculations as required.
1. Display the first 5 rows of the dataset.
2. Display the last 5 rows of the dataset.
3. Get the shape of the dataset (number of rows and columns).
4. Get a summary of the dataset (using .info()).
5. Get basic statistics (mean, standard deviation, etc.) of the dataset using .describe().
6. Check for missing values and display the count of missing values for each column.
7. Fill missing values in the 'Age' column with the median age.
8. Fill missing values in the 'Embarked' column with the most frequent value (mode).
9. Drop the 'Cabin' column due to many missing values.
10. Create a new column, 'FamilySize' by adding the 'SibSp' and 'Parch' columns.
The Titanic dataset contains columns as shown below,
PassengerId
Survived
Pclass
Name
Sex
Age
SibSp
Parch
Ticket
Fare
Cabin
Embarked
Sample Test Cases
Explorer
titanicDat...
Submit
Debugger
import pandas as pd
import numpy as np
# Load the Titanic dataset
data = pd.read_csv('Titanic-Dataset.csv')
# 1. Display the first 5 rows of the dataset
print(data.head(5))
# 2. Display the last 5 rows of the dataset
print(data.tail(5))
# 3. Get the shape of the dataset
Average time
0.160 s
160.00 ms
Maximum time
0.160 s
160.00 ms
1 out of 1 shown test case(s) passed
Test case 1
160 ms
Debug
Expected output
PassengerId Survived Pclass ... Fare Cabin Embarked
0 1 0 3 ... 7.2500 NaN S
1 2 1 1 ... 71.2833 C85 C
2 3 1 3 ... 7.9250 NaN S
Actual output
PassengerId Survived Pclass ... Fare Cabin Embarked
0 1 0 3 ... 7.2500 NaN S
1 2 1 1 ... 71.2833 C85 C
2 3 1 3 ... 7.9250 NaN S
Terminal
Test cases
Prev
Reset
Submit
Next
Search
ENG
IN
13:56
07-05-2025

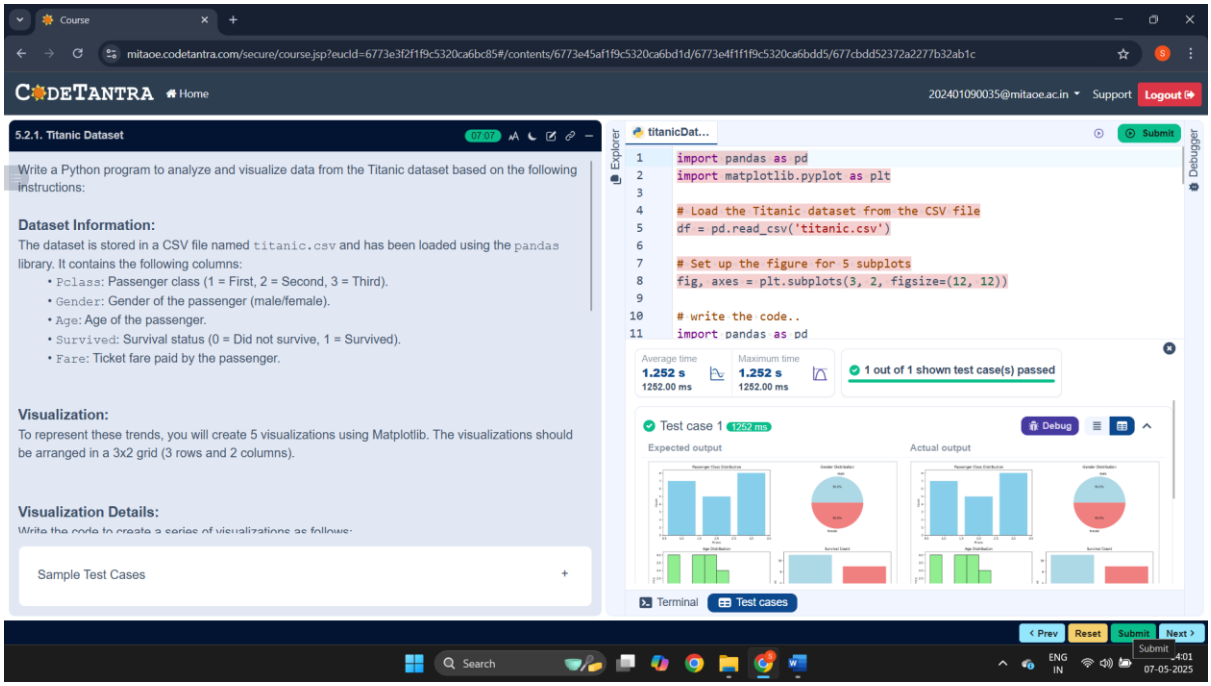

Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e455f1f9c5320ca6bd19/6773e4e4f1f9c5320ca6bdce/67e378a17028471d94dad0a6
CODETANTRA
Home
202401090035@mitaoe.ac.in
Support
Logout
4.2.6. Titanic Dataset Analysis and Data Cleaning - 2
09:54
You are provided with the Titanic dataset containing information about passengers on the Titanic. Your task is to write Python code to answer the following questions based on the dataset.
1. Create a new column 'IsAlone' which is 1 if the passenger is alone (FamilySize = 0), otherwise 0.
2. Convert the 'Sex' column to numeric values (male: 0, female: 1).
3. One-hot encode the 'Embarked' column, dropping the first category.
4. Get the mean age of passengers.
5. Get the median fare of passengers.
6. Get the number of passengers by class.
7. Get the number of passengers by gender.
8. Get the number of passengers by survival status.
9. Calculate the survival rate of passengers.
10. Calculate the survival rate by gender.
The Titanic dataset contains columns as shown below,
Passengerid
Survived
Pclass
Name
Sex
Age
SibSp
Parch
Ticket
Fare
Cabin
Embarked
Sample Test Cases
titanicDat...
Submit
Explorer
Debugger
import pandas as pd
import numpy as np
# Load the Titanic dataset
data = pd.read_csv('Titanic-Dataset.csv')
data['FamilySize'] = data['SibSp'] + data['Parch']
1. Create a new column 'IsAlone' (1 if alone, 0 otherwise)
data['IsAlone']=(data['FamilySize']==0).astype(int)
2. Convert 'Sex' to numeric (male: 0, female: 1)
data['Sex']=data['Sex'].map({'male':0,'female':1})
Average time
0.078 s
78.00 ms
Maximum time
0.078 s
78.00 ms
1 out of 1 shown test case(s) passed
Test case 1
78 ms
Debug
Expected output
Actual output
29.69911764705882
14.4542
3 491
1 216
2 184
Name: Pclass, dtype: int64
Terminal
Test cases
Prev
Reset
Submit
Next
Search
ENG
IN
07-05-2025


Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e455f1f9c5320ca6bd19/6773e4e4f1f9c5320ca6bdce/67e3805d7028471d94dae951
CODETANTRA
Home
202401090035@mitaoe.ac.in
Support
Logout
4.2.7. Titanic Dataset Analysis and Data Cleaning - 3
09:25
You are provided with the Titanic dataset containing information about passengers on the Titanic. Your task is to write Python code to answer the following questions based on the dataset.
1. Calculate the survival rate by class.
2. Calculate the survival rate by embarkation location (Embarked_S).
3. Calculate the survival rate by family size (FamilySize).
4. Calculate the survival rate by being alone (IsAlone).
5. Get the average fare by passenger class (Pclass).
6. Get the average age by passenger class (Pclass).
7. Get the average age by survival status (Survived).
8. Get the average fare by survival status (Survived).
9. Get the number of survivors by class (Pclass).
10. Get the number of non-survivors by class (Pclass).
The Titanic dataset contains columns as shown below,
Passengerid
Survived
Pclass
Name
Sex
Age
SibSp
Parch
Ticket
Fare
Cabin
Embarked
Sample Test Cases
titanicDat...
Submit
Explorer
Debugger
import pandas as pd
import numpy as np
# Load the Titanic dataset
data = pd.read_csv('Titanic-Dataset.csv')
data['FamilySize'] = data['SibSp'] + data['Parch']
data['IsAlone'] = np.where(data['FamilySize'] > 0, 0, 1)
data = pd.get_dummies(data, columns=['Embarked'], drop_first=True)
# 1. Calculate the survival rate by class
survival_by_class=data.groupby('Pclass')['Survived'].mean()
Average time
0.085 s
85.00 ms
Maximum time
0.085 s
85.00 ms
1 out of 1 shown test case(s) passed
Test case 1
85 ms
Debug
Expected output
Actual output
Pclass
1 0.629630
2 0.472826
3 0.242363
Name: Survived, dtype: float64
Embarked_S
Terminal
Test cases
Prev
Reset
Submit
Next
Search
ENG
IN
13:56
07-05-2025


Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e455f1f9c5320ca6bd19/6773e4e4f1f9c5320ca6bdce/67e387fc7028471d94db0238
CODETANTRA
Home
202401090035@mitaoe.ac.in
Support
Logout
4.2.8. Titanic Dataset Analysis and Data Cleaning - 4
06:58
You are provided with the Titanic dataset containing information about passengers on the Titanic. Your task is to write Python code to answer the following questions based on the dataset.
1. Get the number of survivors by gender (Sex).
2. Get the number of non-survivors by gender (Sex).
3. Get the number of survivors by embarkation location (Embarked_S).
4. Get the number of non-survivors by embarkation location (Embarked_S).
5. Calculate the percentage of children (Age < 18) who survived.
6. Calculate the percentage of adults (Age >= 18) who survived.
7. Get the median age of survivors.
8. Get the median age of non-survivors.
9. Get the median fare of survivors.
10. Get the median fare of non-survivors.
The Titanic dataset contains columns as shown below,
Passengerld
Survived
Pclass
Name
Sex
Age
SibSp
Parch
Ticket
Fare
Cabin
Embarked
Sample Test Cases
titanicDat...
Submit
Explorer
Debugger
import pandas as pd
import numpy as np
# Load the Titanic dataset
data = pd.read_csv('Titanic-Dataset.csv')
data = pd.get_dummies(data, columns=['Embarked'], drop_first=True)
# 1. Get the number of survivors by gender
survivors_by_gender=data[data['Survived']==1]['Sex'].value_counts()
print(survivors_by_gender)
Average time
0.077 s
77.00 ms
Maximum time
0.077 s
77.00 ms
1 out of 1 shown test case(s) passed
Test case 1
77 ms
Debug
Expected output
Actual output
female 233
male 109
Name: Sex, dtype: int64
male 468
female 81
Name: Sex, dtype: int64
Terminal
Test cases
Prev
Reset
Submit
Next
Search
ENG
IN
13:57
07-05-2025

Practical 5


Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e45af1f9c5320ca6bd1d/6773e4e9f1f9c5320ca6bdd2/675a8d2c09f9fa022faa9dc0
CODETANTRA
Home
202401090035@mitaoe.ac.in
Support
Logout
5.1.1. Stacked Plot
08:13
Create a stacked area plot to visualize the temperature variations for three different cities (City A, City B, and City C) across the months of the year. The temperature data is provided for each city in the editor.
Your task is to:
• Create a stacked area plot using the data.
• Label the x-axis as "Month", the y-axis as "Temperature", and provide the title "Temperature Variation" for the plot.
• Display the plot showing the temperature variation for each city throughout the months of the year.
Sample Test Cases
stackedpl...
Submit
Explorer
Debugger
1 import matplotlib.pyplot as plt
2 import pandas as pd
3
4 # Data for Months and Temperature for three cities
5 data = {
6 'Month': ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'October', 'November', 'December'],
7 'City_A_Temperature': [5, 7, 10, 13, 17, 20, 22, 21, 18, 12, 8, 6],
8 'City_B_Temperature': [2, 3, 5, 6, 10, 14, 16, 17, 12, 9, 5, 3],
9 'City_C_Temperature': [3, 4, 6, 8, 9, 12, 15, 14, 10, 7, 4, 2]
10 }
Average time
0.320 s
320.00 ms
Maximum time
0.320 s
320.00 ms
1 out of 1 shown test case(s) passed
Test case 1
320 ms
Debug
Expected output
Actual output
Temperature Variation
Terminal
Test cases
Prev
Reset
Submit
Next
Search
ENG
IN
14:00
07-05-2025


Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e45af1f9c5320ca6bd1d/6773e4f1f1f9c5320ca6bdd5/677cbdd52372a2277b32ab1c
CODETANTRA
Home
202401090035@mitaoe.ac.in
Support
Logout
5.2.1. Titanic Dataset
07:07
Write a Python program to analyze and visualize data from the Titanic dataset based on the following instructions:
Dataset Information:
The dataset is stored in a CSV file named titanic.csv and has been loaded using the pandas library. It contains the following columns:
• Pclass: Passenger class (1 = First, 2 = Second, 3 = Third).
• Gender: Gender of the passenger (male/female).
• Age: Age of the passenger.
• Survived: Survival status (0 = Did not survive, 1 = Survived).
• Fare: Ticket fare paid by the passenger.
Visualization:
To represent these trends, you will create 5 visualizations using Matplotlib. The visualizations should be arranged in a 3x2 grid (3 rows and 2 columns).
Visualization Details:
Write the code to create a series of visualizations as follows:
Sample Test Cases
titanicDat...
Submit
Explorer
Debugger
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 # Load the Titanic dataset from the CSV file
5 df = pd.read_csv('titanic.csv')
6
7 # Set up the figure for 5 subplots
8 fig, axes = plt.subplots(3, 2, figsize=(12, 12))
9
10 # write the code..
11 import pandas as pd
Average time
1.252 s
1252.00 ms
Maximum time
1.252 s
1252.00 ms
1 out of 1 shown test case(s) passed
Test case 1
1252 ms
Debug
Expected output
Actual output
Terminal
Test cases
Prev
Reset
Submit
Next
Submit
Search
ENG
IN
14:01
07-05-2025

Course

mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e45af1f9c5320ca6bd1d/6773e4f1f1f9c5320ca6bdd5/67e28a397c0b912d860e689c

CODETANTRA Home 202401090035@mitaoe.ac.in Support Logout

5.2.2. Histogram of passenger information of Titanic

04:37

Write a Python code to plot a histogram for the distribution of the 'Age' column from the Titanic dataset. The histogram should display the frequency of different age ranges with the following specifications:

1. Use **30 bins** for the histogram.
2. Set the **edge color** of the bars to **black** (k).
3. Label the x-axis as **'Age'** and the y-axis as **'Frequency'**.
4. Add the title **"Age Distribution"** to the histogram.

The Titanic dataset contains columns as shown below,

Passengerld	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked

Sample Data:

```
PassengerId,Survived,Pclass,Name,Sex,Age,SibSp,Parch,Ticket,Fare,Cabin,Embarked
1,0,3,"Braund, Mr. Owen Harris",male,22,1,0,A/5 21171,7.25,,S
2,1,1,"Cumings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,PC 17599,71.2833,C85,C
```

Sample Test Cases +

Explorer

Histogra...

Submit

```
1  import pandas as pd
2  import matplotlib.pyplot as plt
3
4  # Load the Titanic dataset
5  data = pd.read_csv('Titanic-Dataset.csv')
6
7  # Data Cleaning
8  data['Age'].fillna(data['Age'].median(), inplace=True)
9  data['Embarked'].fillna(data['Embarked'].mode()[0], inplace=True)
10 data.drop('Cabin', axis=1, inplace=True)
11
```

Debugger

Average time 0.307 s 307.00 ms | Maximum time 0.307 s 307.00 ms | 1 out of 1 shown test case(s) passed

Test case 1 307 ms Debug

Expected output

Actual output

Terminal Test cases

‹ Prev Reset Submit Next ›

Search ENG IN 14:01 07-05-2025

Course

mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e45af1f9c5320ca6bd1d/6773e4f1f1f9c5320ca6bdd5/67e28d6e7c0b912d860e75ad

CODETANTRA Home 202401090035@mitaoe.ac.in Support Logout

5.2.3. Bar plot of survival rate of passengers

03:39

Write a Python code to plot a bar chart that shows the count of passengers who survived and did not survive in the Titanic dataset. The chart should display the following specifications:

1. Use the '**Survived**' column to show the count of survivors (0 = Did not survive, 1 = Survived).
2. Set the chart type to '**bar**'.
3. Add the title "**Survival Count**" to the chart.
4. Label the x-axis as '**Survived**' and the y-axis as '**Count**'.

The Titanic dataset contains columns as shown below,

Passengerld	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked

Sample Data:

```
PassengerId,Survived,Pclass,Name,Sex,Age,SibSp,Parch,Ticket,Fare,Cabin,Embarked
1,0,3,"Braund, Mr. Owen Harris",male,22,1,0,A/5 21171,7.25,,S
2,1,1,"Cumings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,PC 17599,71.2833,C85,C
```

Sample Test Cases +

Explorer

BarPlotOf...

Submit

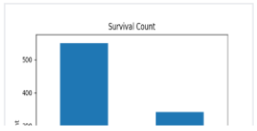
```
1  import pandas as pd
2  import matplotlib.pyplot as plt
3
4  # Load the Titanic dataset
5  data = pd.read_csv('Titanic-Dataset.csv')
6
7  # Data Cleaning
8  data['Age'].fillna(data['Age'].median(), inplace=True)
9  data['Embarked'].fillna(data['Embarked'].mode()[0], inplace=True)
10 data.drop('Cabin', axis=1, inplace=True)
11
```

Debugger

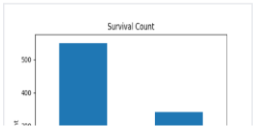
Average time 0.365 s 365.00 ms | Maximum time 0.365 s 365.00 ms | 1 out of 1 shown test case(s) passed

Test case 1 365 ms Debug

Expected output



Actual output



Terminal Test cases

‹ Prev Reset Submit Next ›

Search ENG IN 14:01 07-05-2025

Course

mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e45af1f9c5320ca6bd1d/6773e4f1f1f9c5320ca6bdd5/67e28fb27c0b912d860e7db4

CODETANTRA Home 202401090035@mitaoe.ac.in Support Logout

5.2.4. Bar Plot for Survival by Gender

01:56

Write a Python code to plot a stacked bar chart that shows the count of passengers who survived and did not survive, grouped by gender, in the Titanic dataset. The chart should display the following specifications:

1. Group the data by the **'Sex'** column, then use the **value_counts()** function to count the occurrences of survivors (0 = Did not survive, 1 = Survived) for each gender.
2. Use a **stacked bar chart** to display the survival counts.
3. Add the title **"Survival by Gender"** to the chart.
4. Label the x-axis as **'Gender'** and the y-axis as **'Count'**.
5. The legend should indicate **'Not Survived'** and **'Survived'**.

The Titanic dataset contains columns as shown below,

Passengerid	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked

Sample Data:

Sample Test Cases +

Explorer

BarPlotOf...

Submit

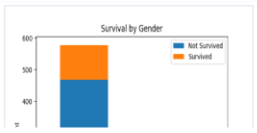
Debugger

```
1  import pandas as pd
2  import matplotlib.pyplot as plt
3
4  # Load the Titanic dataset
5  data = pd.read_csv('Titanic-Dataset.csv')
6
7  # Data Cleaning
8  data['Age'].fillna(data['Age'].median(), inplace=True)
9  data['Embarked'].fillna(data['Embarked'].mode()[0], inplace=True)
10 data.drop('Cabin', axis=1, inplace=True)
11
```

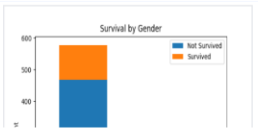
Average time 0.407 s 407.00 ms | Maximum time 0.407 s 407.00 ms | 1 out of 1 shown test case(s) passed

Test case 1 407 ms Debug

Expected output



Actual output



Terminal Test cases

‹ Prev Reset Submit Next ›

Search ENG IN 14:02 07-05-2025

Course

mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e45af1f9c5320ca6bd1d/6773e4f1f1f9c5320ca6bdd5/67e291be7c0b912d860e876f

CODETANTRA Home 202401090035@mitaoe.ac.in Support Logout

5.2.5. Bar Plot for Survival by Pclass

02:30

Write a Python code to plot a stacked bar chart that shows the count of passengers who survived and did not survive, grouped by passenger class (**Pclass**), in the Titanic dataset. The chart should display the following specifications:

1. Group the data by the **Pclass** column and count the number of survivors (0 = Did not survive, 1 = Survived) for each class using **value_counts()**.
2. Use a **stacked bar chart** to display the survival counts.
3. Add the title **"Survival by Pclass"** to the chart.
4. Label the x-axis as **'Pclass'** and the y-axis as **'Count'**.
5. The legend should indicate **'Not Survived'** and **'Survived'**.

The Titanic dataset contains columns as shown below,

Passengerid	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked

Sample Data:

Sample Test Cases +

Explorer

BarPlotOf...

Submit

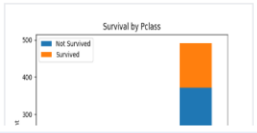
Debugger

```
1  import pandas as pd
2  import matplotlib.pyplot as plt
3
4  # Load the Titanic dataset
5  data = pd.read_csv('Titanic-Dataset.csv')
6
7  # Data Cleaning
8  data['Age'].fillna(data['Age'].median(), inplace=True)
9  data['Embarked'].fillna(data['Embarked'].mode()[0], inplace=True)
10 data.drop('Cabin', axis=1, inplace=True)
11
```

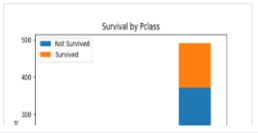
Average time 0.434 s 434.00 ms | Maximum time 0.434 s 434.00 ms | 1 out of 1 shown test case(s) passed

Test case 1 434 ms Debug

Expected output



Actual output



Terminal Test cases

‹ Prev Reset Submit Next ›

Search ENG IN 4:02 07-05-2025

Course

mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e45af1f9c5320ca6bd1d/6773e4f1f1f9c5320ca6bdd5/67e297bf7c0b912d860e9c81

CODETANTRA Home 202401090035@mitaoe.ac.in Support Logout

5.2.6. Bar Plot for Survival by Embarked

02:12

Write a Python code to plot a stacked bar chart showing the survival count for passengers based on their embarkation location in the Titanic dataset.
The chart should display the following specifications:

1. Use the **Embarked** column to determine the embarkation location. After converting this column into dummy variables (using **pd.get_dummies()**), plot the survival count based on the **Embarked_Q** column (representing passengers who embarked from Queenstown) in relation to survival.
2. Set the chart type to 'bar' and make it stacked.
3. Add the title **"Survival by Embarked "** to the chart.
4. Label the x-axis as **'Embarked'** and the y-axis as **'Count'**.
5. Include a legend to distinguish between survivors and non-survivors (label the legend as **'Survived'** and **'Not Survived'**).

The Titanic dataset contains columns as shown below,

Passengerld	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked

Sample Test Cases +

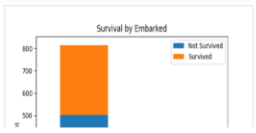
Explorer BarPlotOf... Submit Debugger

```
1  import pandas as pd
2  import matplotlib.pyplot as plt
3
4  # Load the Titanic dataset
5  data = pd.read_csv('Titanic-Dataset.csv')
6
7  # Data Cleaning
8  data['Age'].fillna(data['Age'].median(), inplace=True)
9  data['Embarked'].fillna(data['Embarked'].mode()[0], inplace=True)
10 data.drop('Cabin', axis=1, inplace=True)
11
```

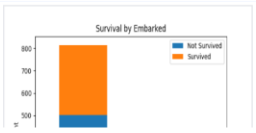
Average time 0.406 s 406.00 ms | Maximum time 0.406 s 406.00 ms | 1 out of 1 shown test case(s) passed

Test case 1 406 ms Debug

Expected output



Actual output



Terminal Test cases

‹ Prev Reset Submit Next ›

Search ENG IN 03 07-05-2025

Course

mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e45af1f9c5320ca6bd1d/6773e4f1f1f9c5320ca6bdd5/67e2992e7c0b912d860ea071

CODETANTRA Home 202401090035@mitaoe.ac.in Support Logout

5.2.7. Box plot for Age Distribution

03:15

Write a Python code to plot a boxplot that shows the distribution of the 'Age' column from the Titanic dataset across different passenger classes. The boxplot should display the following specifications:

1. Use the **Pclass** column to group the data for the boxplot.
2. Set the title of the plot to **"Age by Pclass"**.
3. Remove the default subtitle with **plt.suptitle('')**.
4. Label the x-axis as **'Pclass'** and the y-axis as **'Age'**.

The Titanic dataset contains columns as shown below,

Passengerld	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked

Sample Data:

```
PassengerId,Survived,Pclass,Name,Sex,Age,SibSp,Parch,Ticket,Fare,Cabin,Embarked
1,0,3,"Braund, Mr. Owen Harris",male,22,1,0,A/5 21171,7.25,,S
2,1,1,"Cumings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,PC 17599,71.2833,C85,C
```

Sample Test Cases +

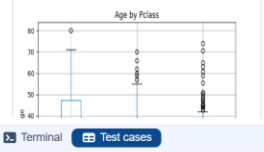
Explorer BoxPlotF... Submit Debugger

```
1  import pandas as pd
2  import matplotlib.pyplot as plt
3
4  # Load the Titanic dataset
5  data = pd.read_csv('Titanic-Dataset.csv')
6
7  # Data Cleaning
8  data['Age'].fillna(data['Age'].median(), inplace=True)
9  data['Embarked'].fillna(data['Embarked'].mode()[0], inplace=True)
10 data.drop('Cabin', axis=1, inplace=True)
11
```

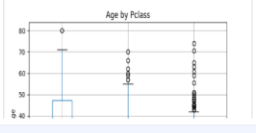
Average time 0.405 s 405.00 ms | Maximum time 0.405 s 405.00 ms | 1 out of 1 shown test case(s) passed

Test case 1 405 ms Debug

Expected output



Actual output



Terminal Test cases

‹ Prev Reset Submit Next ›

Search ENG IN 14:03 07-05-2025

Course

mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e45af1f9c5320ca6bd1d/6773e4f1f1f9c5320ca6bdd5/67e29db57c0b912d860eab45

CODETANTRA Home 202401090035@mitaoe.ac.in Support Logout

5.2.8. Box Plot for Age by Survived

02:17

Write a Python code to plot a boxplot that shows the distribution of the 'Age' column from the Titanic dataset based on whether passengers survived or not. The boxplot should display the following specifications:

1. Use the **Survived** column to group the data for the boxplot (0 = Did not survive, 1 = Survived).
2. Set the title of the plot to **"Age by Survival"**.
3. Remove the default subtitle with **plt.suptitle('')**.
4. Label the x-axis as **'Survived'** and the y-axis as **'Age'**.

The Titanic dataset contains columns as shown below,

Passengerld	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked

Sample Data:

```
PassengerId,Survived,Pclass,Name,Sex,Age,SibSp,Parch,Ticket,Fare,Cabin,Embarked
1,0,3,"Braund, Mr. Owen Harris",male,22,1,0,A/5 21171,7.25,,S
```

Sample Test Cases +

Explorer

BoxPlotF...

Submit

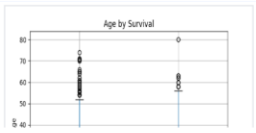
```
1  import pandas as pd
2  import matplotlib.pyplot as plt
3
4  # Load the Titanic dataset
5  data = pd.read_csv('Titanic-Dataset.csv')
6
7  # Data Cleaning
8  data['Age'].fillna(data['Age'].median(), inplace=True)
9  data['Embarked'].fillna(data['Embarked'].mode()[0], inplace=True)
10 data.drop('Cabin', axis=1, inplace=True)
11
```

Debugger

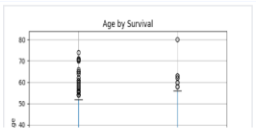
Average time 0.349 s 349.00 ms | Maximum time 0.349 s 349.00 ms | 1 out of 1 shown test case(s) passed

Test case 1 349 ms Debug

Expected output



Actual output



Terminal Test cases

Prev Reset Submit Next

Search ENG IN 14:03 07-05-2025

Course

mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e45af1f9c5320ca6bd1d/6773e4f1f1f9c5320ca6bdd5/67e2a1727c0b912d860eb39b

CODETANTRA Home 202401090035@mitaoe.ac.in Support Logout

5.2.9. Box Plot for Fare by Pclass

02:46

Write a Python code to plot a boxplot that shows the distribution of the 'Fare' column from the Titanic dataset based on the passenger class (Pclass). The boxplot should display the following specifications:

1. Use the **Pclass** column to group the data for the boxplot.
2. Set the title of the plot to **"Fare by Pclass"**.
3. Remove the default subtitle with **plt.suptitle('')**.
4. Label the x-axis as **'Pclass'** and the y-axis as **'Fare'**.

The Titanic dataset contains columns as shown below,

Passengerld	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked

Sample Data:

```
PassengerId,Survived,Pclass,Name,Sex,Age,SibSp,Parch,Ticket,Fare,Cabin,Embarked
1,0,3,"Braund, Mr. Owen Harris",male,22,1,0,A/5 21171,7.25,,S
2,1,1,"Cumings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,PC 17599,71.2833,C85,C
```

Sample Test Cases +

Explorer

BoxPlotF...

Submit

```
1  import pandas as pd
2  import matplotlib.pyplot as plt
3
4  # Load the Titanic dataset
5  data = pd.read_csv('Titanic-Dataset.csv')
6
7  # Data Cleaning
8  data['Age'].fillna(data['Age'].median(), inplace=True)
9  data['Embarked'].fillna(data['Embarked'].mode()[0], inplace=True)
10 data.drop('Cabin', axis=1, inplace=True)
11
```

Debugger

Average time 0.333 s 333.00 ms | Maximum time 0.333 s 333.00 ms | 1 out of 1 shown test case(s) passed

Test case 1 333 ms Debug

Expected output

Actual output

Terminal Test cases

Prev Reset Submit Next

Search ENG IN 14:04 07-05-2025

Course

mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e45af1f9c5320ca6bd1d/6773e4f1f1f9c5320ca6bdd5/67e2a2977c0b912d860eb621

CODETANTRA Home

202401090035@mitaoe.ac.in Support Logout

5.2.10. Scatter Plot for Age vs. Fare

02:14

Write a Python code to plot a scatter plot showing the relationship between the 'Age' and 'Fare' columns in the Titanic dataset. The scatter plot should display the following specifications:

1. Use the **Age** column for the x-axis and the **Fare** column for the y-axis.
2. Set the title of the plot to **"Age vs. Fare"**.
3. Label the x-axis as **'Age'** and the y-axis as **'Fare'**.

The Titanic dataset contains columns as shown below,

Passengerid	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked

Sample Data:

```
PassengerId,Survived,Pclass,Name,Sex,Age,SibSp,Parch,Ticket,Fare,Cabin,Embarked
1,0,3,"Braund, Mr. Owen Harris",male,22,1,0,A/5 21171,7.25,,S
2,1,1,"Cumings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,PC 17599,71.2833,C85,C
3,1,3,"Heikkinen, Miss. Laina",female,26,0,0,STON/O2. 3101282,7.925,,S
```

Sample Test Cases +

Explorer

AgeFareS...

Submit

Debugger

```
1  import pandas as pd
2  import matplotlib.pyplot as plt
3
4  # Load the Titanic dataset
5  data = pd.read_csv('Titanic-Dataset.csv')
6
7  # Data Cleaning
8  data['Age'].fillna(data['Age'].median(), inplace=True)
9  data['Embarked'].fillna(data['Embarked'].mode()[0], inplace=True)
10 data.drop('Cabin', axis=1, inplace=True)
11
```

Average time 0.293 s 293.00 ms | Maximum time 0.293 s 293.00 ms | 1 out of 1 shown test case(s) passed

Test case 1 293 ms | Debug

Expected output

Actual output

Terminal | Test cases

‹ Prev | Reset | Submit | Next ›

Search

ENG IN 14:04 07-05-2025

Course

mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e45af1f9c5320ca6bd1d/6773e4f1f1f9c5320ca6bdd5/67e2a5387c0b912d860eba29

CODETANTRA Home

202401090035@mitaoe.ac.in Support Logout

5.2.11. Scatter Plot for Age vs. Fare by Survived

02:34

Write a Python code to plot a scatter plot showing the relationship between the 'Age' and 'Fare' columns in the Titanic dataset, with points color-coded by survival status. The scatter plot should display the following specifications:

1. Use the **Age** column for the x-axis and the **Fare** column for the y-axis.
2. Color the points based on the **Survived** column: **Red** for passengers who did not survive (**Survived = 0**). **Blue** for passengers who survived (**Survived = 1**).
3. Set the title of the plot to **"Age vs. Fare by Survival"**.
4. Label the x-axis as **'Age'** and the y-axis as **'Fare'**.

The Titanic dataset contains columns as shown below,

Passengerid	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked

Sample Data:

Sample Test Cases +

Explorer

AgeFareS...

Submit

Debugger

```
1  import pandas as pd
2  import matplotlib.pyplot as plt
3
4  # Load the Titanic dataset
5  data = pd.read_csv('Titanic-Dataset.csv')
6
7  # Data Cleaning
8  data['Age'].fillna(data['Age'].median(), inplace=True)
9  data['Embarked'].fillna(data['Embarked'].mode()[0], inplace=True)
10 data.drop('Cabin', axis=1, inplace=True)
11
```

Average time 0.316 s 316.00 ms | Maximum time 0.316 s 316.00 ms | 1 out of 1 shown test case(s) passed

Test case 1 316 ms | Debug

Expected output

Actual output

Terminal | Test cases

‹ Prev | Reset | Submit

Search

ENG IN 14:04 07-05-2025