Course

https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e447f1f9c5320ca6bce4/6773e494f1f9c5320ca6bd76/6773e9c2f4ab787eca9d09e4

CODETANTRA Home 202401040349@mitaoe.ac.in Support Logout

## 1.1.1. Calculate Momentum

07:00

Write a program that accepts the mass of an object (in kilograms) and its velocity (in meters per second), then calculates and displays the momentum of the object. The momentum $p$ is calculated using the formula:

$$p = m \times v$$

where:
$m$ is the mass of the object (in kilograms).
$v$ is the velocity of the object (in meters per second).

**Input Format:**
A single floating-point number representing the mass of the object in kilograms.
A single floating-point number representing the velocity of the object in meters per second.

**Output Format:**
The output will display calculated momentum with appropriate units (kgm/s) (rounded up to 2 decimal places).

Sample Test Cases +

Explorer

calculate...

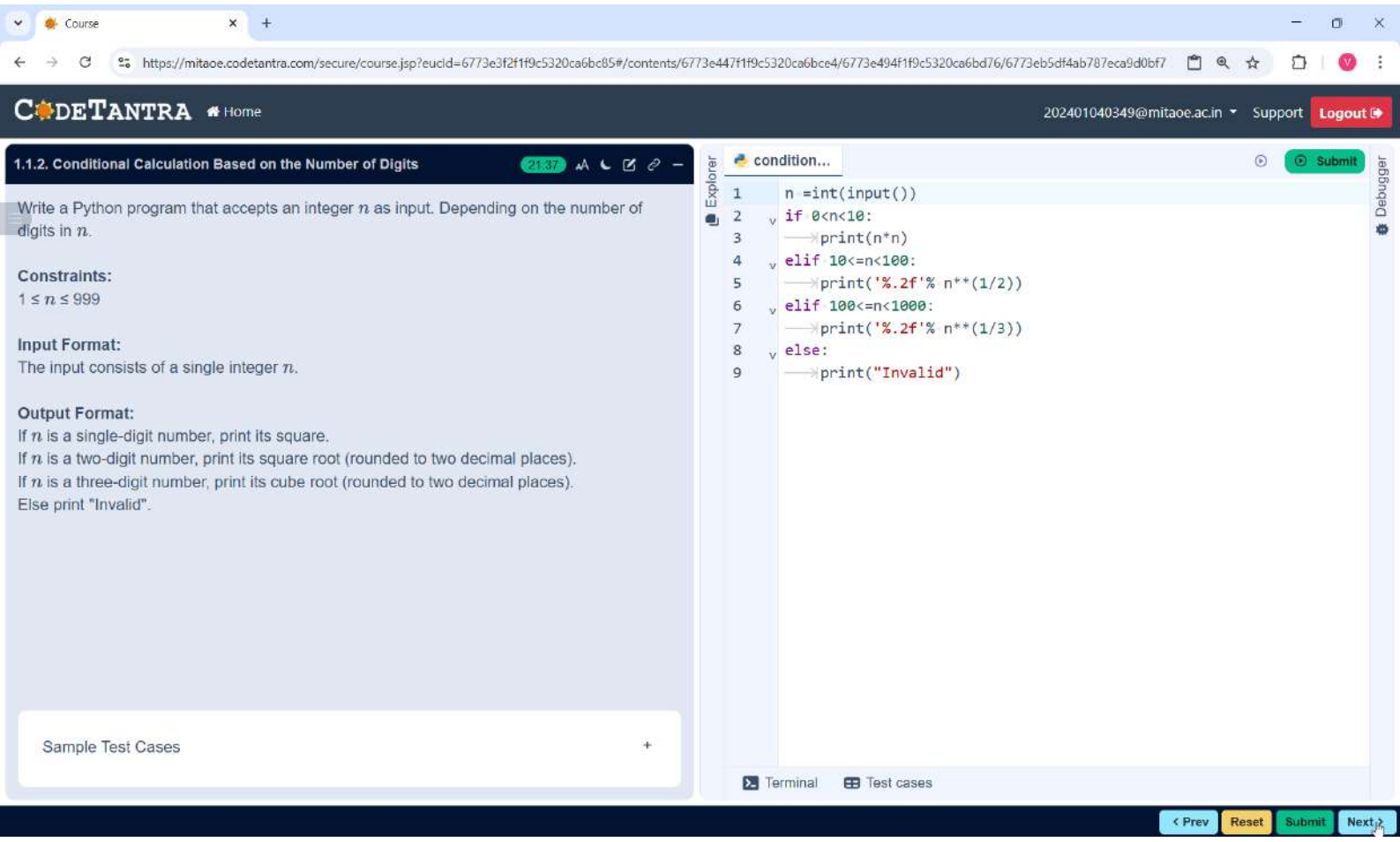
Submit

Debugger

```
1  m = float(input())
2  v = float(input())
3  p= m*v
4  print('%.2fkgm/s'%p)
```

Terminal Test cases

< Prev Reset Submit Next >


Course
https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e447f1f9c5320ca6bce4/6773e494f1f9c5320ca6bd76/6773eb5df4ab787eca9d0bf7
CODETANTRA
Home
202401040349@mitaoe.ac.in
Support
Logout
1.1.2. Conditional Calculation Based on the Number of Digits
21:37
Write a Python program that accepts an integer $n$ as input. Depending on the number of digits in $n$.
Constraints:
$1 \leq n \leq 999$
Input Format:
The input consists of a single integer $n$.
Output Format:
If $n$ is a single-digit number, print its square.
If $n$ is a two-digit number, print its square root (rounded to two decimal places).
If $n$ is a three-digit number, print its cube root (rounded to two decimal places).
Else print "Invalid".
Sample Test Cases
Explorer
condition...
Submit
Debugger
1 n =int(input())
2 if 0<n<10:
3 print(n*n)
4 elif 10<=n<100:
5 print('%.2f'% n**(1/2))
6 elif 100<=n<1000:
7 print('%.2f'% n**(1/3))
8 else:
9 print("Invalid")
Terminal
Test cases
< Prev
Reset
Submit
Next >

Course

https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e447f1f9c5320ca6bce4/6773e494f1f9c5320ca6bd76/6773ef42f4ab787eca9d11f7

CODETANTRA Home

202401040349@mitaoe.ac.in Support Logout

## 1.1.3. Age and Salary Calculation

37:00

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 +

Explorer

birthDate...

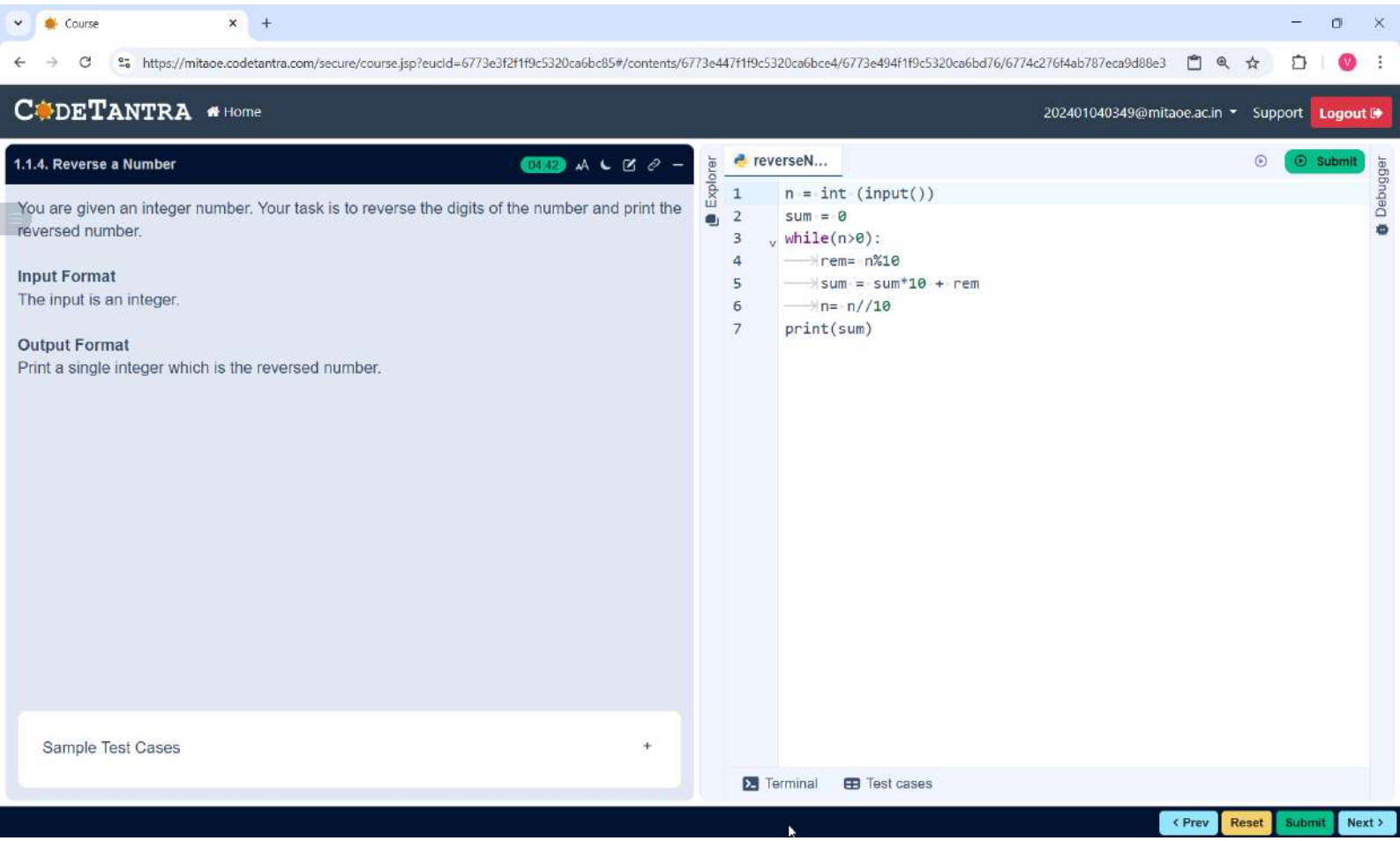
Submit

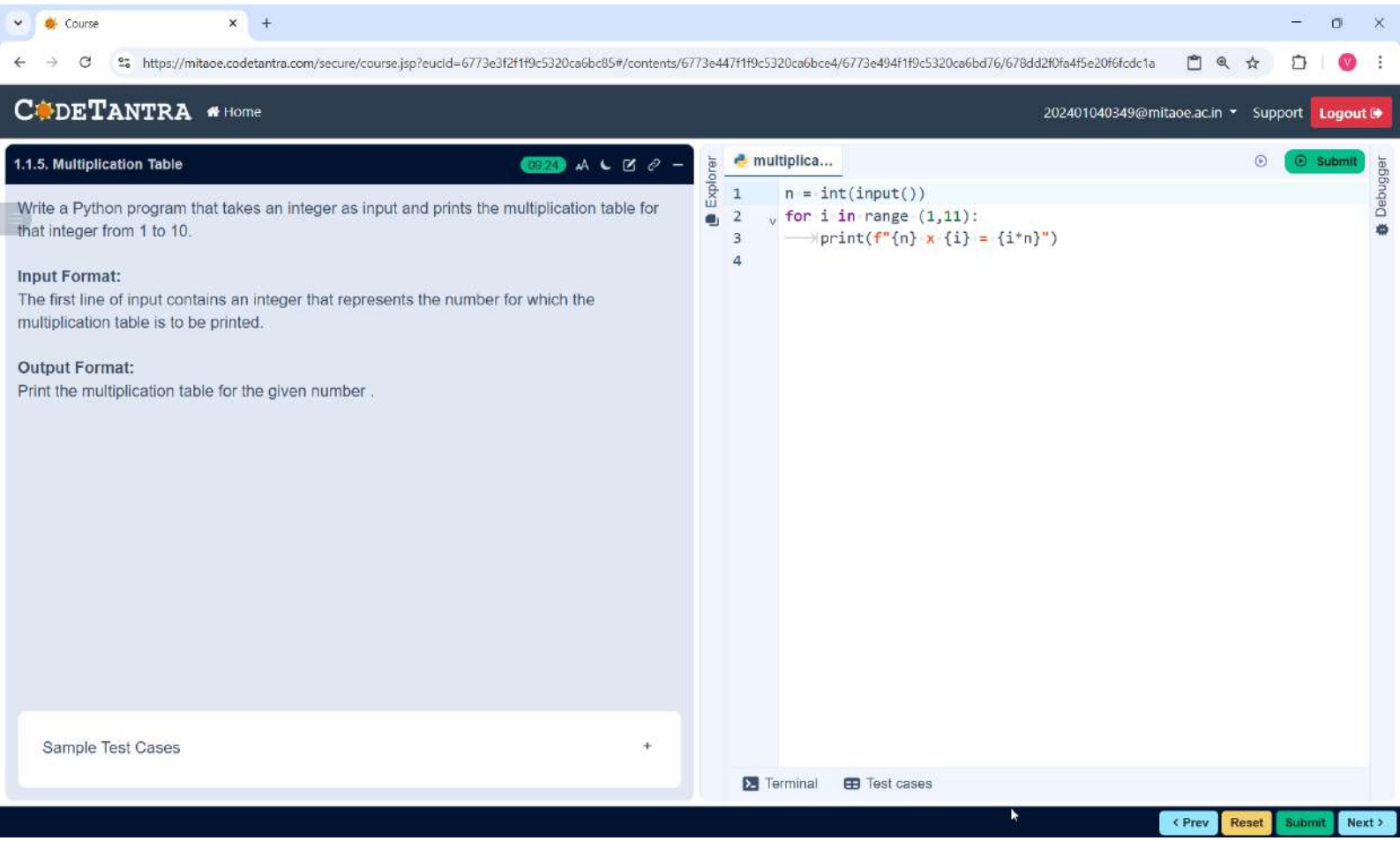
Debugger

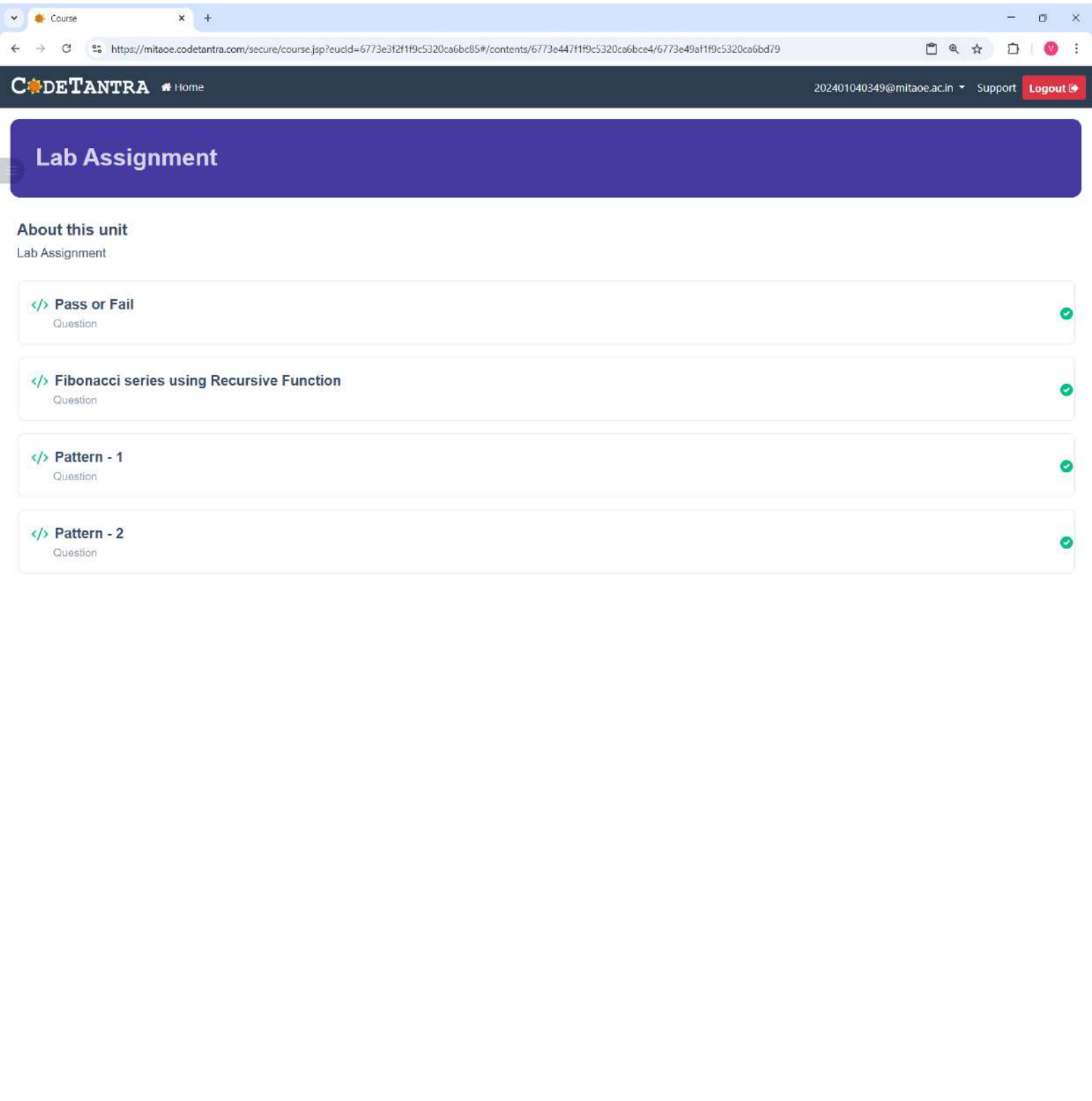
```
1  from datetime import datetime
2
3  def calculate_age(birthdate):
4      date_object = datetime.strptime(birthdate, "%d-%m-%Y")
5      today = datetime.today()
6      if (today.month, today.day) < (date_object.month,
   date_object.day):
7          age = today.year-date_object.year- ((today.month,
   today.day) < (date_object.month, date_object.day))
8          return age
9
10     elif((today.month, today.day) > (date_object.month,
   date_object.day)):
11         age = today.year-date_object.year- ((today.month,
   today.day) > (date_object.month, date_object.day))
12         return age
13
14
15
16 def convert_salary_to_dollars(salary_in_rupees):
17     salary = salary_in_rupees * 0.012
18     return salary
19     ....
20     ....
21 birthdate = input()
22 salary_in_rupees = float(input())
```

Terminal Test cases

< Prev Reset Submit Next >


Course
https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e447f1f9c5320ca6bce4/6773e494f1f9c5320ca6bd76/6774c276f4ab787eca9d88e3
CODETANTRA
Home
202401040349@mitaoe.ac.in
Support
Logout
1.1.4. Reverse a Number
04:42
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
Explorer
reverseN...
Submit
Debugger
1 n = int (input())
2 sum = 0
3 while(n>0):
4 rem= n%10
5 sum = sum*10 + rem
6 n= n//10
7 print(sum)
Terminal
Test cases
Prev
Reset
Submit
Next


Course
https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e447f1f9c5320ca6bce4/6773e494f1f9c5320ca6bd76/678dd2f0fa4f5e20f6fcdc1a
CODETANTRA
Home
202401040349@mitaoe.ac.in
Support
Logout
1.1.5. Multiplication Table
09:24
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
Explorer
multiplica...
Submit
Debugger
1 n = int(input())
2 for i in range (1,11):
3 print(f"{n} x {i} = {i*n}")
4
Terminal
Test cases
Prev
Reset
Submit
Next


Course
https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e447f1f9c5320ca6bce4/6773e49af1f9c5320ca6bd79
CODETANTRA
Home
202401040349@mitaoe.ac.in
Support
Logout
Lab Assignment
About this unit
Lab Assignment
Pass or Fail
Question
Fibonacci series using Recursive Function
Question
Pattern - 1
Question
Pattern - 2
Question

Course

https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e447f1f9c5320ca6bce4/6773e49af1f9c5320ca6bd79/6773f211f4ab787eca9d15d0

CODETANTRA Home 202401040349@mitaoe.ac.in Support Logout

## 1.2.1. Pass or Fail

32.48

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:

- "Fail".

Sample Test Cases +

Explorer

passorFa...

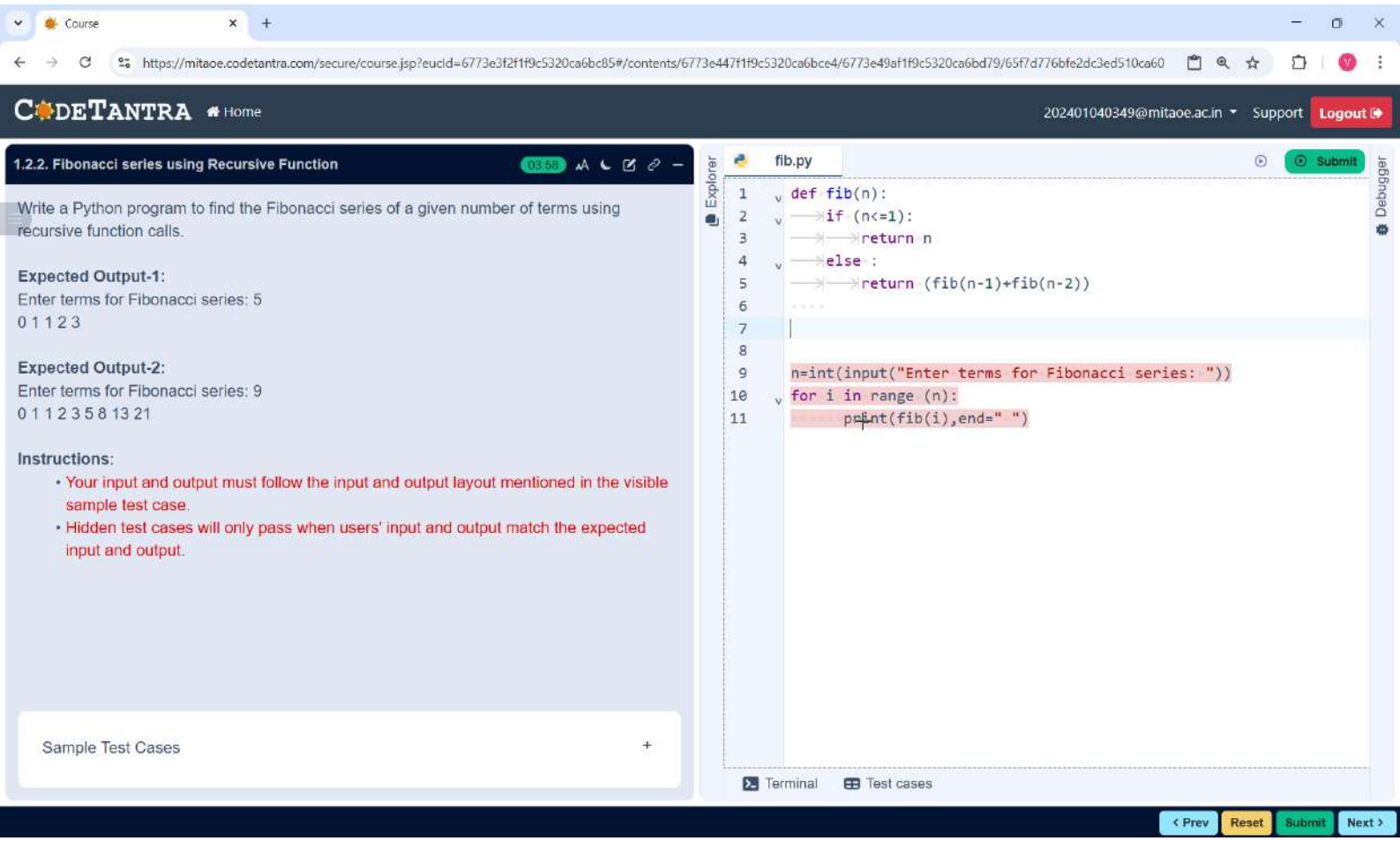
Submit

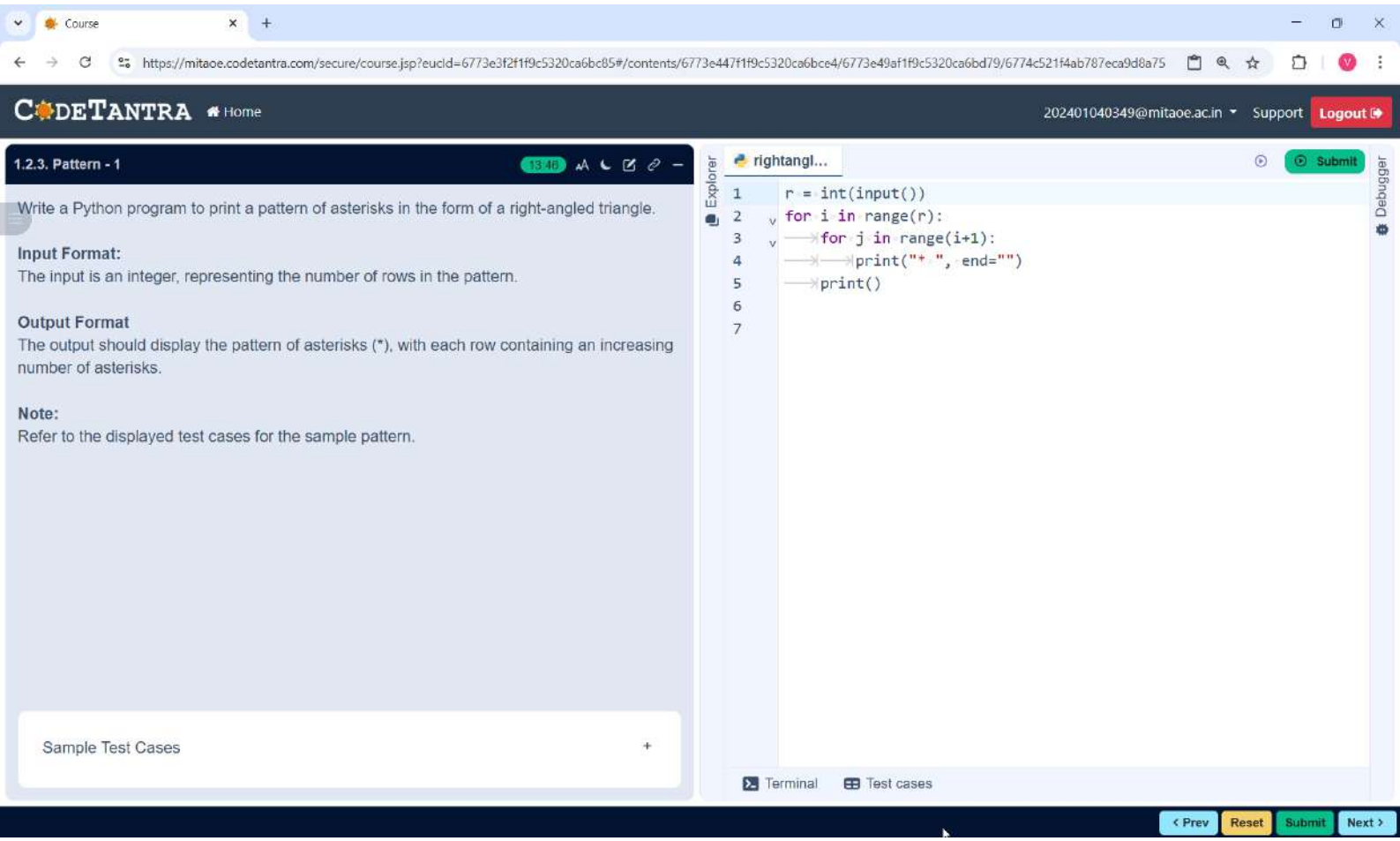
Debugger

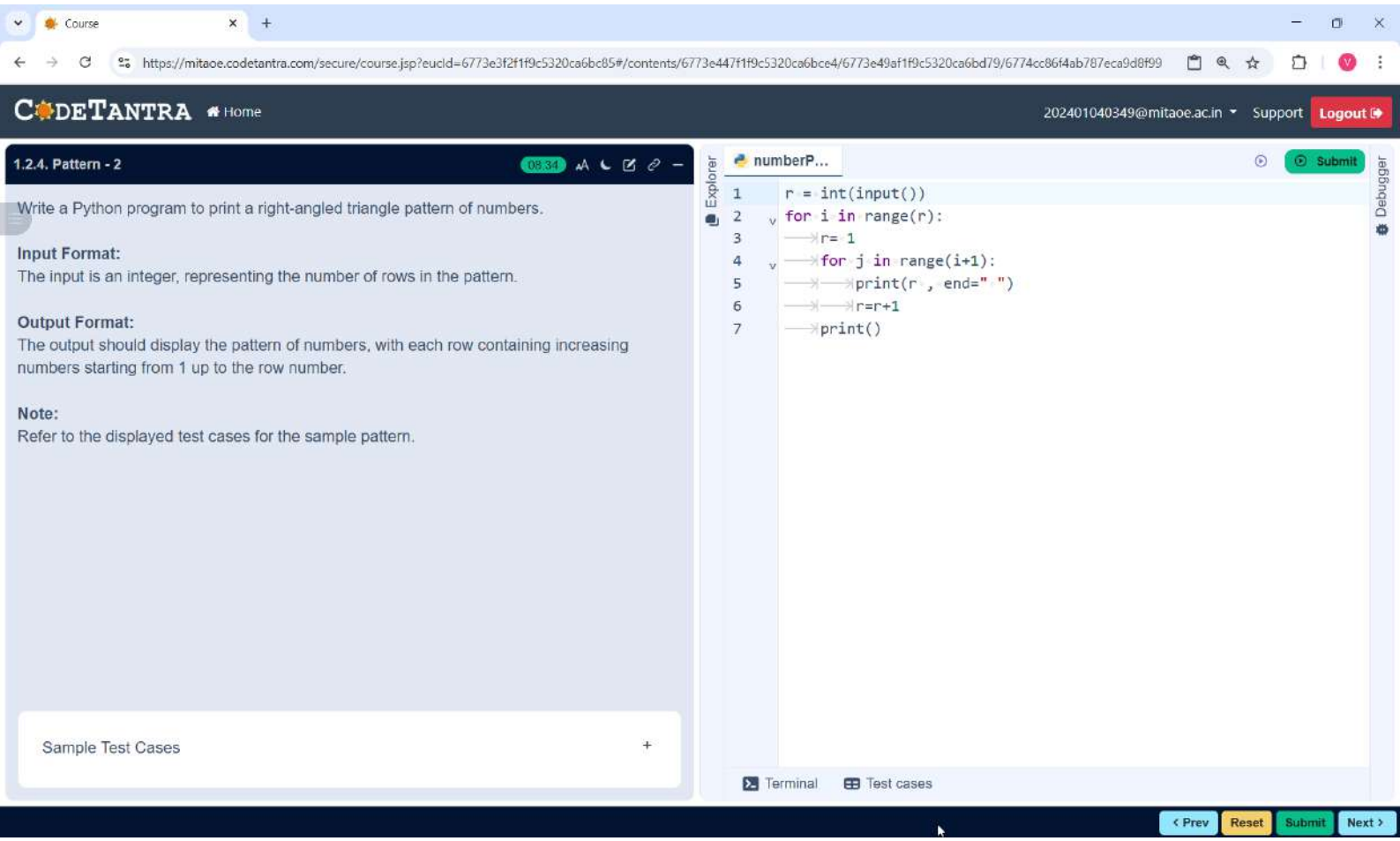
```
1  n = int(input())
2  marks = list(map(int,input().split()))
3  if all(mark>= 40 for mark in marks):
4      aggregate_percent = sum(marks)/n
5      print(f"Aggregate Percentage:{aggregate_percent: .2f}")
6      if(aggregate_percent >= 75):
7          print("Grade: Distinction")
8      elif(60<= aggregate_percent < 75):
9          print("Grade: First Division")
10     elif(50 <= aggregate_percent <60) :
11         print("Grade: Second Division")
12     elif (40 <= aggregate_percent <50) :
13         print("Grade: Third Division")
14 else:
15     print("Fail")
16
```

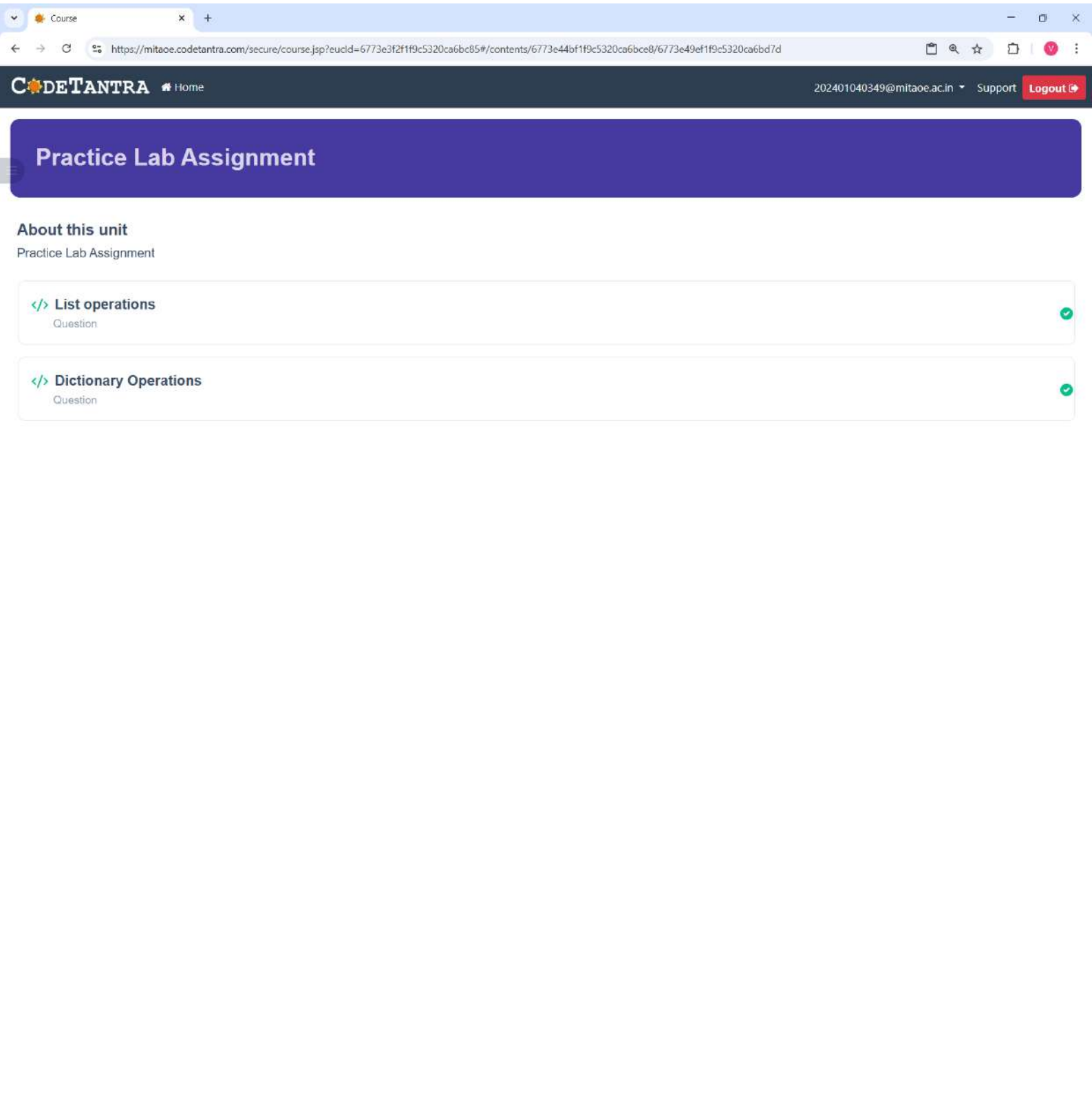
Terminal Test cases

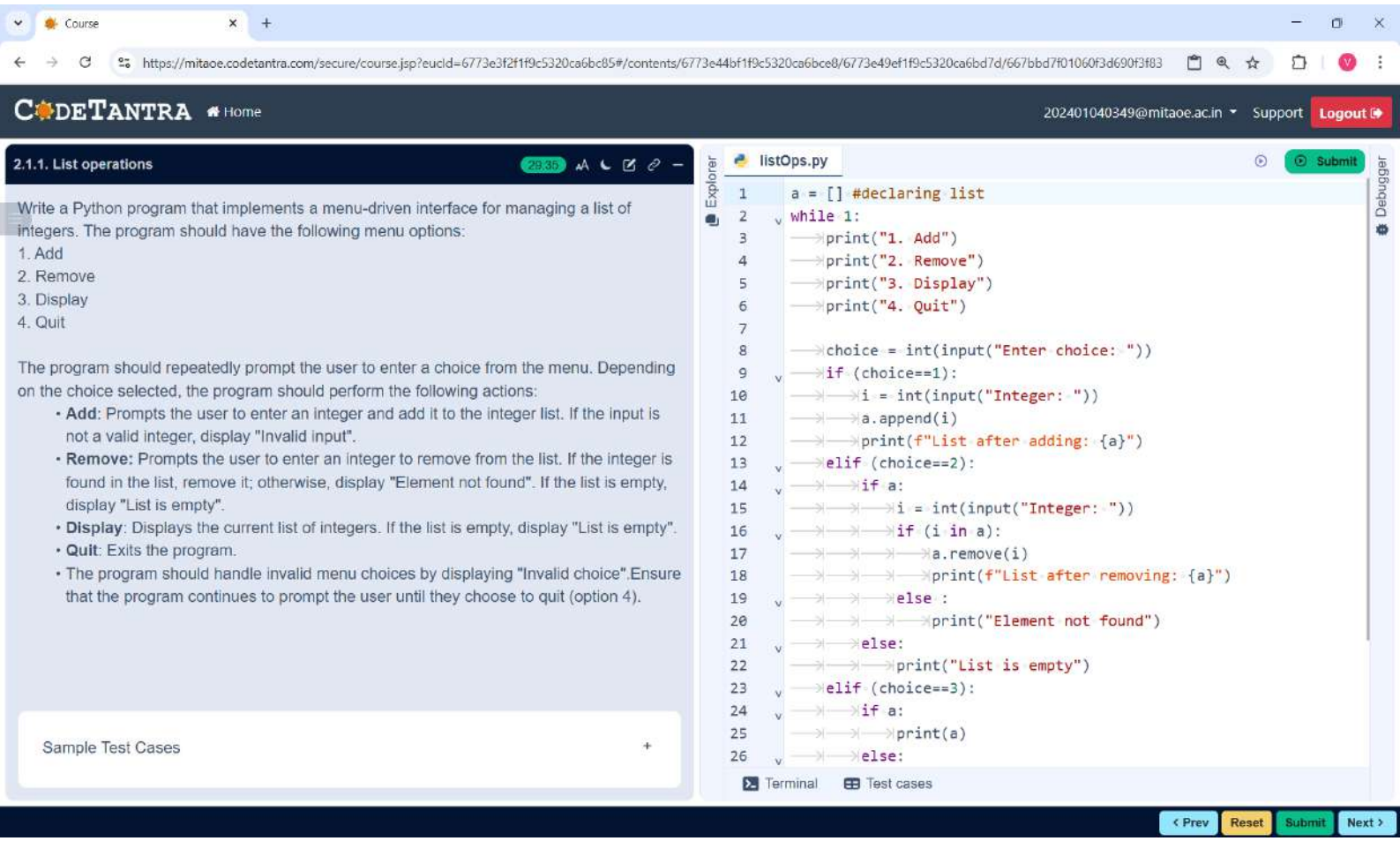
< Prev Reset Submit Next >

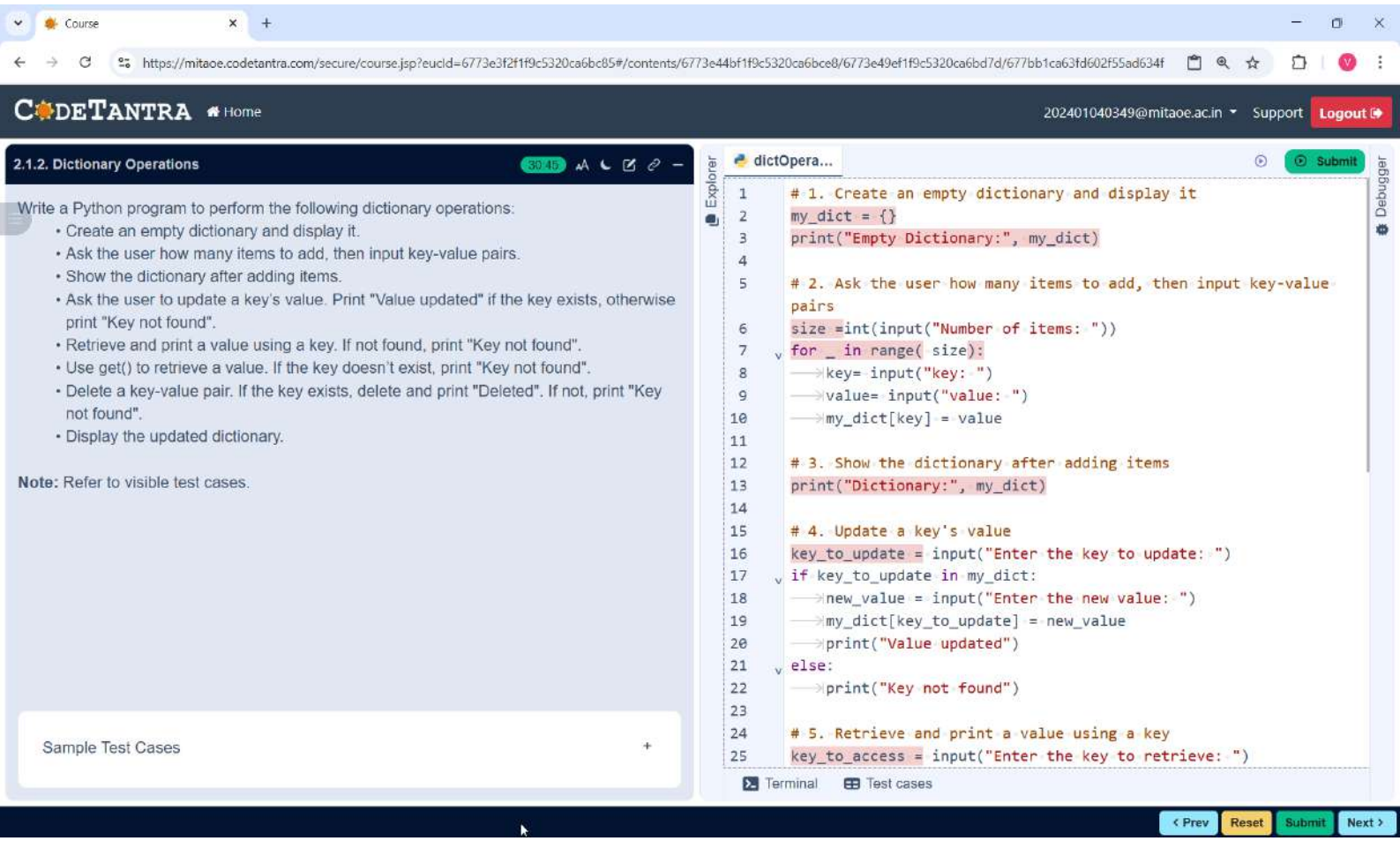

Course
https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e447f1f9c5320ca6bce4/6773e49af1f9c5320ca6bd79/65f7d776bfe2dc3ed510ca60
CODETANTRA
Home
202401040349@mitaoe.ac.in
Support
Logout
1.2.2. Fibonacci series using Recursive Function
03:58
Write a Python program 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
+
Explorer
fib.py
Submit
Debugger
1 def fib(n):
2 if (n<=1):
3 return n
4 else :
5 return (fib(n-1)+fib(n-2))
6
7
8
9 n=int(input("Enter terms for Fibonacci series: "))
10 for i in range (n):
11 print(fib(i),end=" ")
Terminal
Test cases
‹ Prev
Reset
Submit
Next ›


Course
https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e447f1f9c5320ca6bce4/6773e49af1f9c5320ca6bd79/6774c521f4ab787eca9d8a75
CODETANTRA
Home
202401040349@mitaoe.ac.in
Support
Logout
1.2.3. Pattern - 1
13:46
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
Explorer
rightangl...
Submit
Debugger
1 r = int(input())
2 for i in range(r):
3 for j in range(i+1):
4 print("* ", end="")
5 print()
6
7
Terminal
Test cases
Prev
Reset
Submit
Next


Course
https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e447f1f9c5320ca6bce4/6773e49af1f9c5320ca6bd79/6774cc86f4ab787eca9d8f99
CODETANTRA
Home
202401040349@mitaoe.ac.in
Support
Logout
1.2.4. Pattern - 2
08:34
Write a Python program to print a right-angled triangle pattern of numbers.
Input Format:
The input is an integer, representing the number of rows in the pattern.
Output Format:
The output should display the pattern of numbers, with each row containing increasing numbers starting from 1 up to the row number.
Note:
Refer to the displayed test cases for the sample pattern.
Sample Test Cases
Explorer
numberP...
Submit
Debugger
1 r = int(input())
2 for i in range(r):
3 r= 1
4 for j in range(i+1):
5 print(r , end=" ")
6 r=r+1
7 print()
Terminal
Test cases
Prev
Reset
Submit
Next


Course
https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e44bf1f9c5320ca6bce8/6773e49ef1f9c5320ca6bd7d
CODETANTRA
Home
202401040349@mitaoe.ac.in
Support
Logout
Practice Lab Assignment
About this unit
Practice Lab Assignment
List operations
Question
Dictionary Operations
Question


Course
https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e44bf1f9c5320ca6bce8/6773e49ef1f9c5320ca6bd7d/667bbd7f01060f3d690f3f83
CODETANTRA
Home
202401040349@mitaoe.ac.in
Support
Logout
2.1.1. List operations
29.35
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
a = [] #declaring list
while 1:
print("1. Add")
print("2. Remove")
print("3. Display")
print("4. Quit")
choice = int(input("Enter choice: "))
if (choice==1):
i = int(input("Integer: "))
a.append(i)
print(f"List after adding: {a}")
elif (choice==2):
if a:
i = int(input("Integer: "))
if (i in a):
a.remove(i)
print(f"List after removing: {a}")
else :
print("Element not found")
else:
print("List is empty")
elif (choice==3):
if a:
print(a)
else:
Terminal
Test cases
Prev
Reset
Submit
Next


Course
https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e44bf1f9c5320ca6bce8/6773e49ef1f9c5320ca6bd7d/677bb1ca63fd602f55ad634f
CODETANTRA
Home
202401040349@mitaoe.ac.in
Support
Logout
2.1.2. Dictionary Operations
30:45
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
1 # 1. Create an empty dictionary and display it
2 my_dict = {}
3 print("Empty Dictionary:", my_dict)
4
5 # 2. Ask the user how many items to add, then input key-value pairs
6 size =int(input("Number of items: "))
7 for _ in range( size):
8 key= input("key: ")
9 value= input("value: ")
10 my_dict[key] = value
11
12 # 3. Show the dictionary after adding items
13 print("Dictionary:", my_dict)
14
15 # 4. Update a key's value
16 key_to_update = input("Enter the key to update: ")
17 if key_to_update in my_dict:
18 new_value = input("Enter the new value: ")
19 my_dict[key_to_update] = new_value
20 print("Value updated")
21 else:
22 print("Key not found")
23
24 # 5. Retrieve and print a value using a key
25 key_to_access = input("Enter the key to retrieve: ")
Terminal
Test cases
< Prev
Reset
Submit
Next >

Course

https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e44bf1f9c5320ca6bce8/6773e49ef1f9c5320ca6bd7d/677bb1ca63fd602f55ad634f

CODETANTRA Home

202401040349@mitaoe.ac.in Support Logout

## 2.1.2. Dictionary Operations

30:45

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...

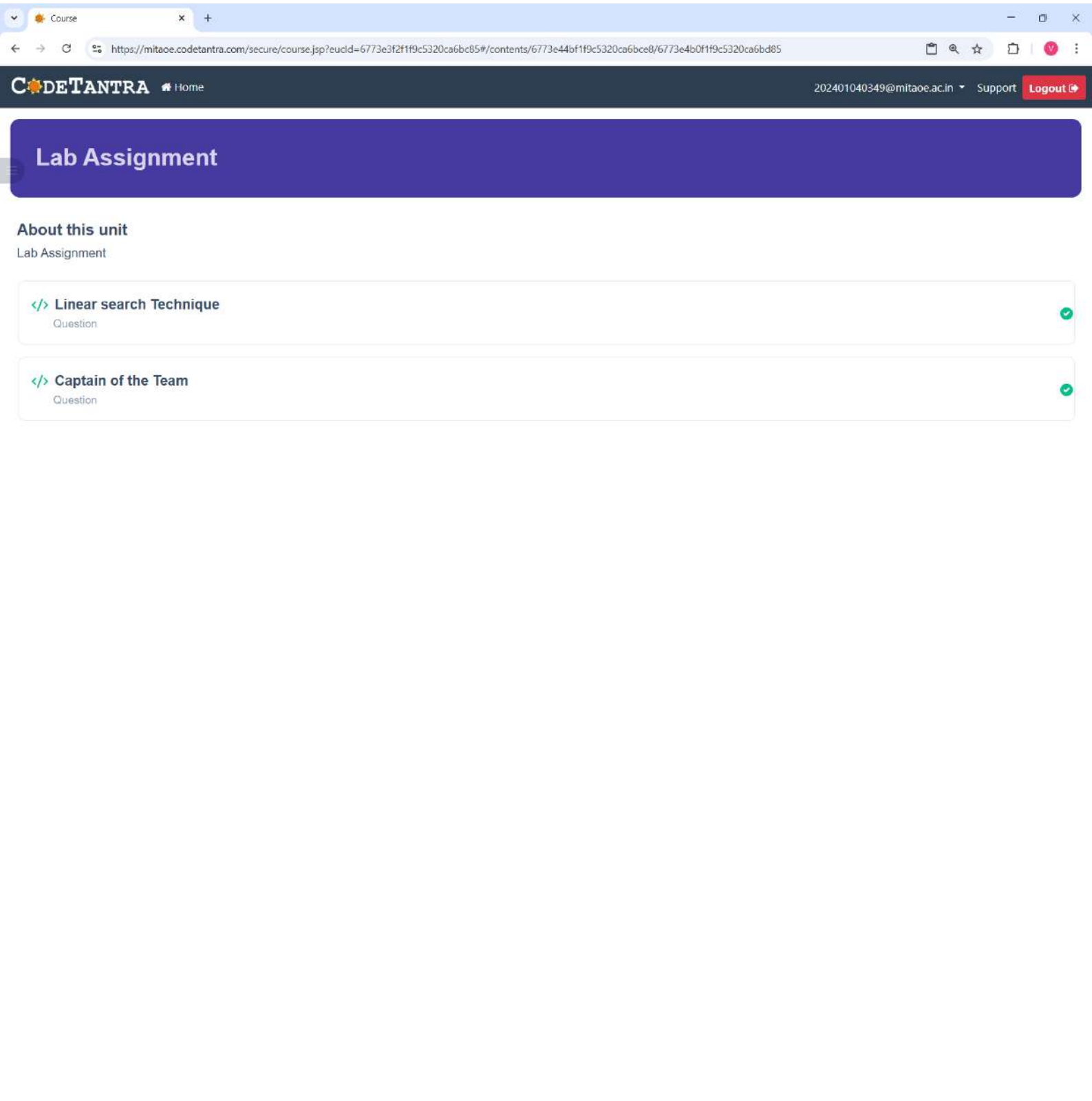
Submit

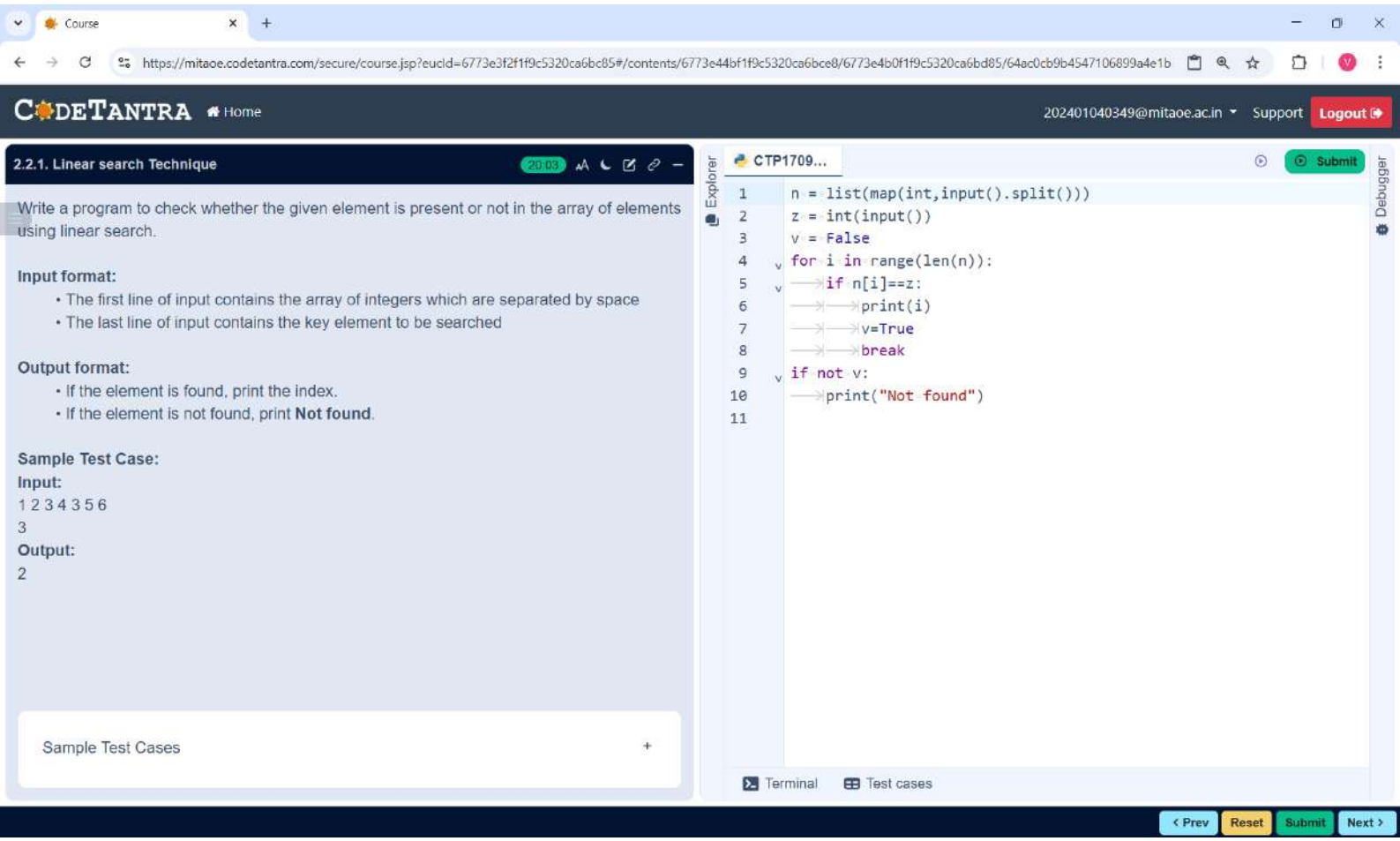
Debugger

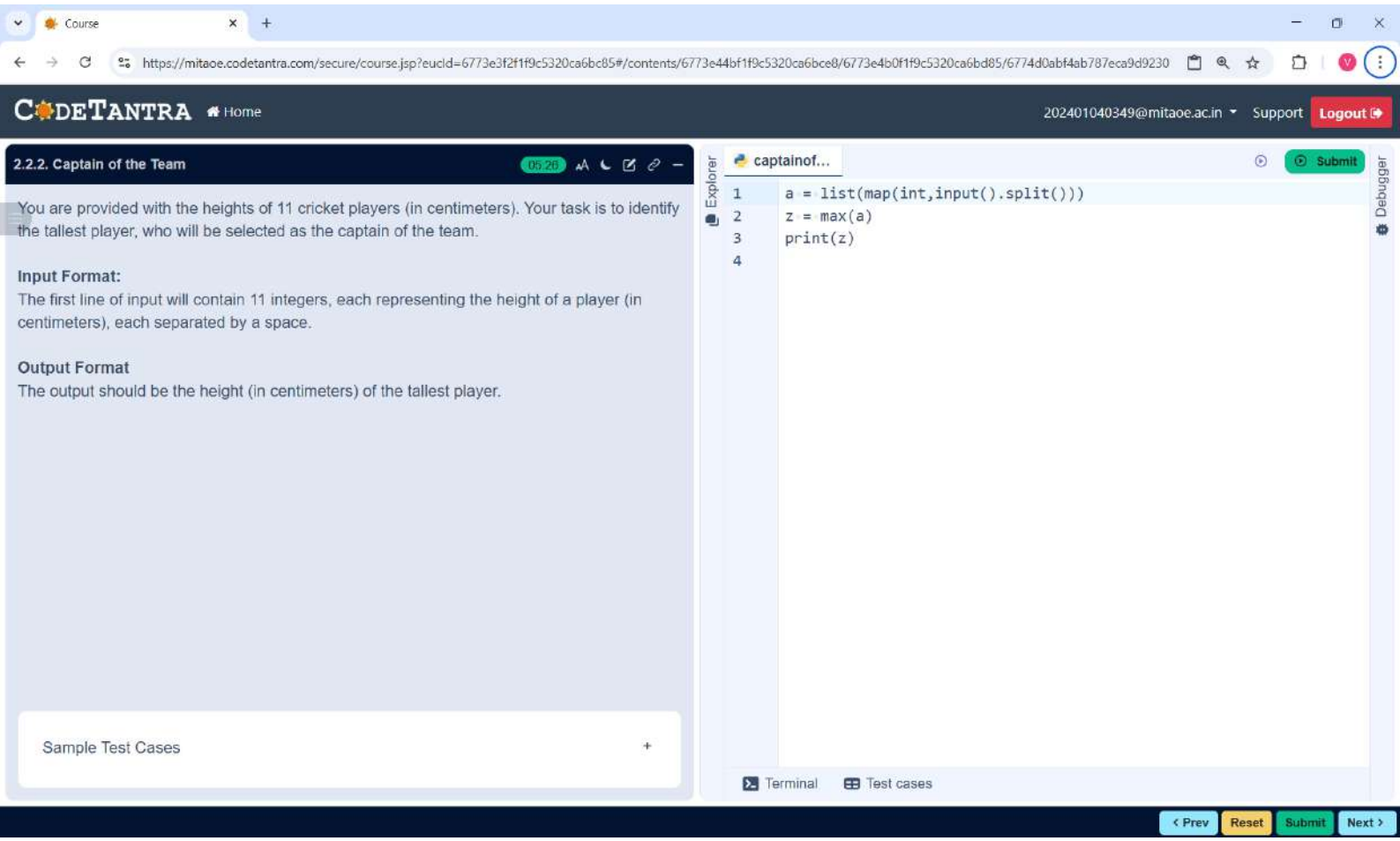
```
25  key_to_access = input("Enter the key to retrieve: ")
26  if key_to_access in my_dict:
27      print(f"Key: {key_to_access}, Value:
    {my_dict[key_to_access]}")
28  else:
29      print("Key not found")
30
31  # 6. Use `get()` to retrieve a value
32  key_to_get = input("Enter the key to get using the get() method:
    ")
33  value = my_dict.get(key_to_get)
34  if value is not None:
35      print(f"Key: {key_to_get}, Value: {value}")
36  else:
37      print("Key not found")
38
39  # 7. Delete a key-value pair
40  key_to_delete = input("Enter the key to delete: ")
41  if key_to_delete in my_dict:
42      del my_dict[key_to_delete]
43      print("Deleted")
44  else:
45      print("Key not found")
46
47  # 8. Display the updated dictionary
48  print("Updated Dictionary:", my_dict)
```

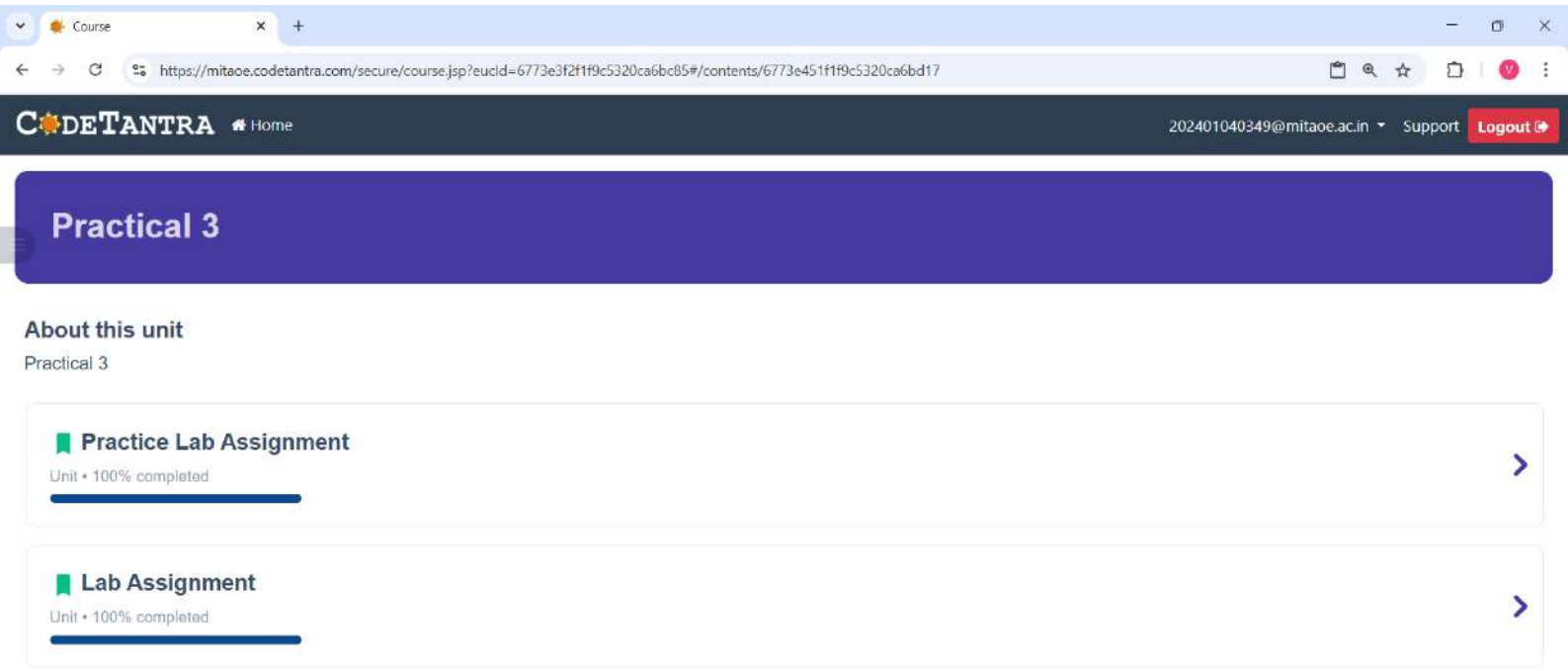
Terminal Test cases

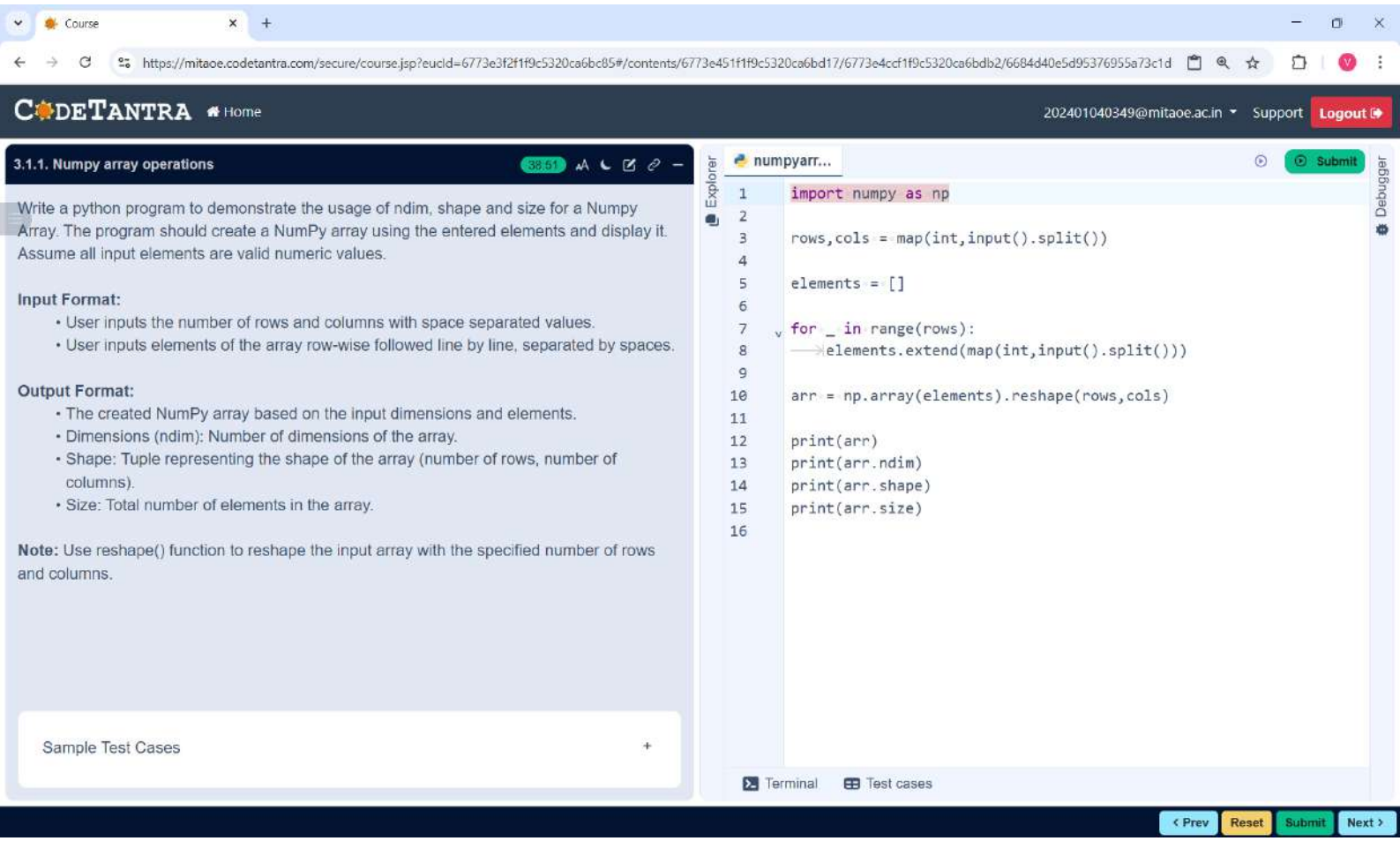
< Prev Reset Submit Next >

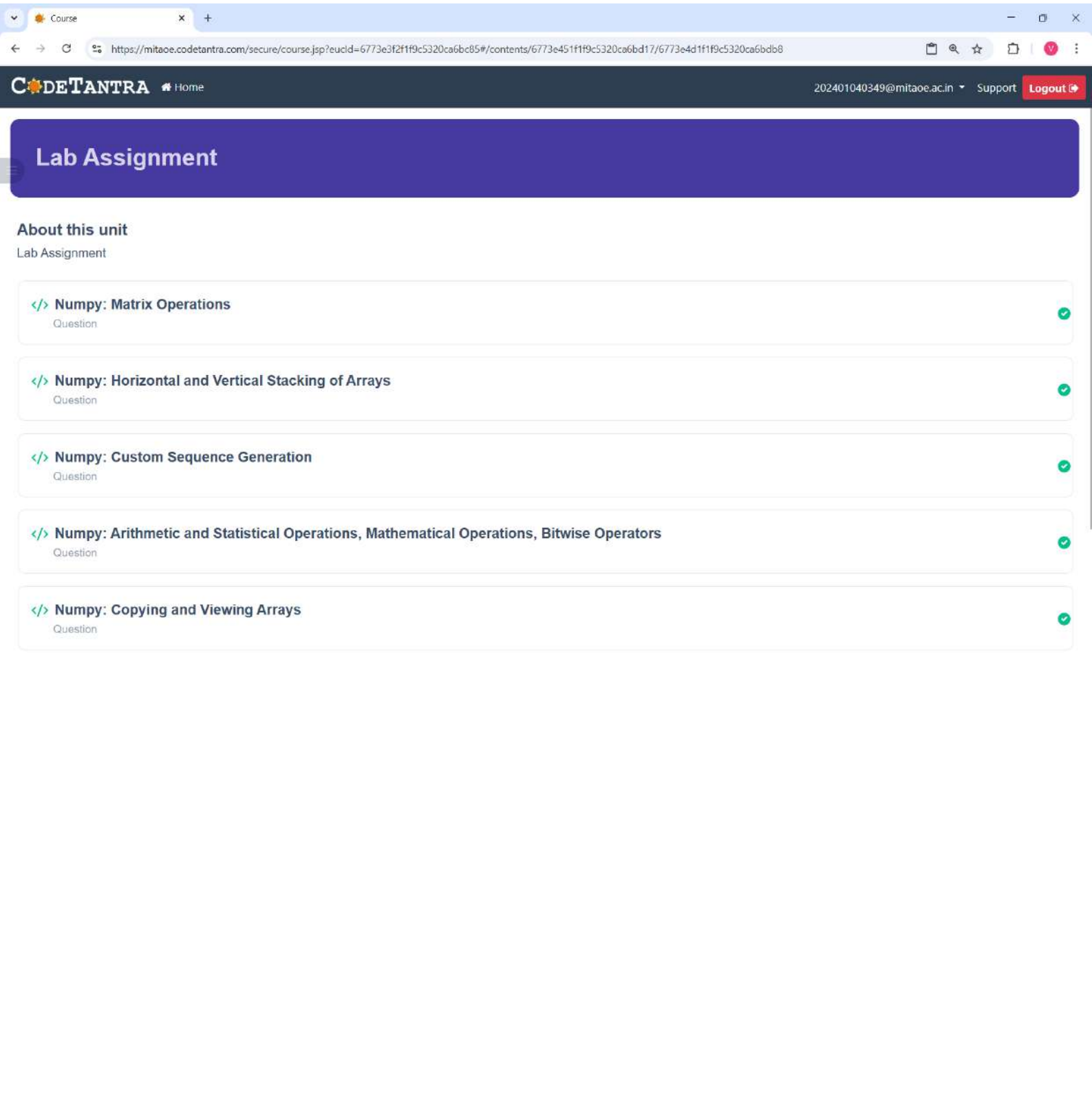

Course
https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e44bf1f9c5320ca6bce8/6773e4b0f1f9c5320ca6bd85
CODETANTRA
Home
202401040349@mitaoe.ac.in
Support
Logout
Lab Assignment
About this unit
Lab Assignment
Linear search Technique
Question
Captain of the Team
Question

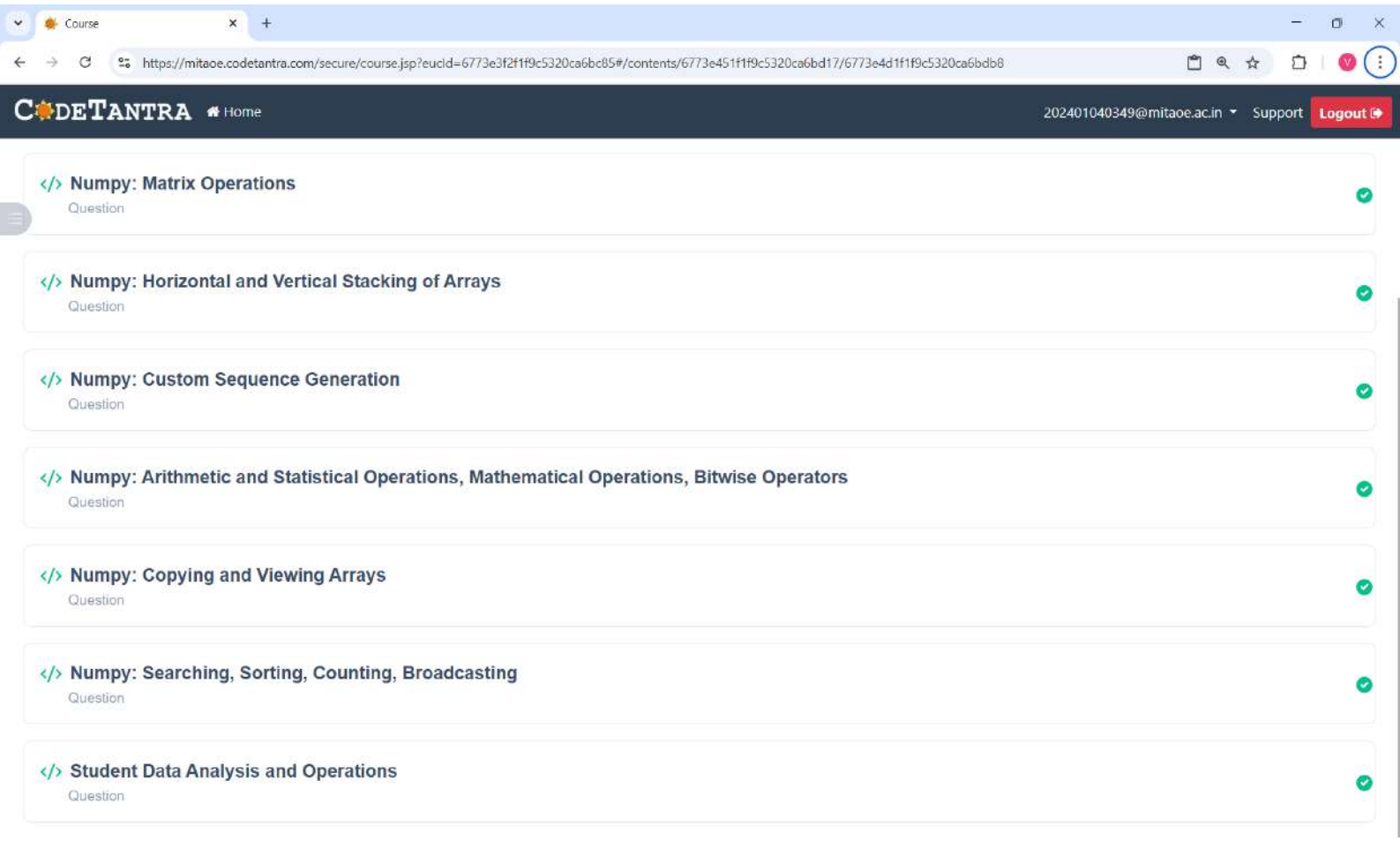

Course
https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e44bf1f9c5320ca6bce8/6773e4b0f1f9c5320ca6bd85/64ac0cb9b4547106899a4e1b
CODETANTRA
Home
202401040349@mitaoe.ac.in
Support
Logout
2.2.1. Linear search Technique
20:03
Write a program to check whether the 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
Explorer
CTP1709...
Submit
Debugger
1 n = list(map(int,input().split()))
2 z = int(input())
3 v = False
4 for i in range(len(n)):
5 if n[i]==z:
6 print(i)
7 v=True
8 break
9 if not v:
10 print("Not found")
11
Terminal
Test cases
Prev
Reset
Submit
Next


Course
https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e44bf1f9c5320ca6bce8/6773e4b0f1f9c5320ca6bd85/6774d0abf4ab787eca9d9230
CODETANTRA
Home
202401040349@mitaoe.ac.in
Support
Logout
2.2.2. Captain of the Team
05:26
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
Explorer
captainof...
Submit
Debugger
1 a = list(map(int,input().split()))
2 z = max(a)
3 print(z)
4
Terminal
Test cases
Prev
Reset
Submit
Next


Course
https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e451f1f9c5320ca6bd17
CODETANTRA
Home
202401040349@mitaoe.ac.in
Support
Logout
Practical 3
About this unit
Practical 3
Practice Lab Assignment
Unit • 100% completed
Lab Assignment
Unit • 100% completed


Course
https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e451f1f9c5320ca6bd17/6773e4ccf1f9c5320ca6bdb2/6684d40e5d95376955a73c1d
CODETANTRA
Home
202401040349@mitaoe.ac.in
Support
Logout
3.1.1. Numpy array operations
38:51
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
1 import numpy as np
2
3 rows,cols = map(int,input().split())
4
5 elements = []
6
7 for _ in range(rows):
8 elements.extend(map(int,input().split()))
9
10 arr = np.array(elements).reshape(rows,cols)
11
12 print(arr)
13 print(arr.ndim)
14 print(arr.shape)
15 print(arr.size)
16
Terminal
Test cases
< Prev
Reset
Submit
Next >


Course
https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e451f1f9c5320ca6bd17/6773e4d1f1f9c5320ca6bdb8
CODETANTRA
Home
202401040349@mitaoe.ac.in
Support
Logout
Lab Assignment
About this unit
Lab Assignment
Numpy: Matrix Operations
Question
Numpy: Horizontal and Vertical Stacking of Arrays
Question
Numpy: Custom Sequence Generation
Question
Numpy: Arithmetic and Statistical Operations, Mathematical Operations, Bitwise Operators
Question
Numpy: Copying and Viewing Arrays
Question


Course
https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e451f1f9c5320ca6bd17/6773e4d1f1f9c5320ca6bdb8
CODETANTRA
Home
202401040349@mitaoe.ac.in
Support
Logout
Numpy: Matrix Operations
Question
Numpy: Horizontal and Vertical Stacking of Arrays
Question
Numpy: Custom Sequence Generation
Question
Numpy: Arithmetic and Statistical Operations, Mathematical Operations, Bitwise Operators
Question
Numpy: Copying and Viewing Arrays
Question
Numpy: Searching, Sorting, Counting, Broadcasting
Question
Student Data Analysis and Operations
Question

Course

https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e451f1f9c5320ca6bd17/6773e4d1f1f9c5320ca6bdb8/6774e3fff4ab787eca9da092

CODETANTRA Home 202401040349@mitaoe.ac.in Support Logout

## 3.2.1. Numpy: Matrix Operations

20:26

The given code takes two $3 \times 3$ 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...

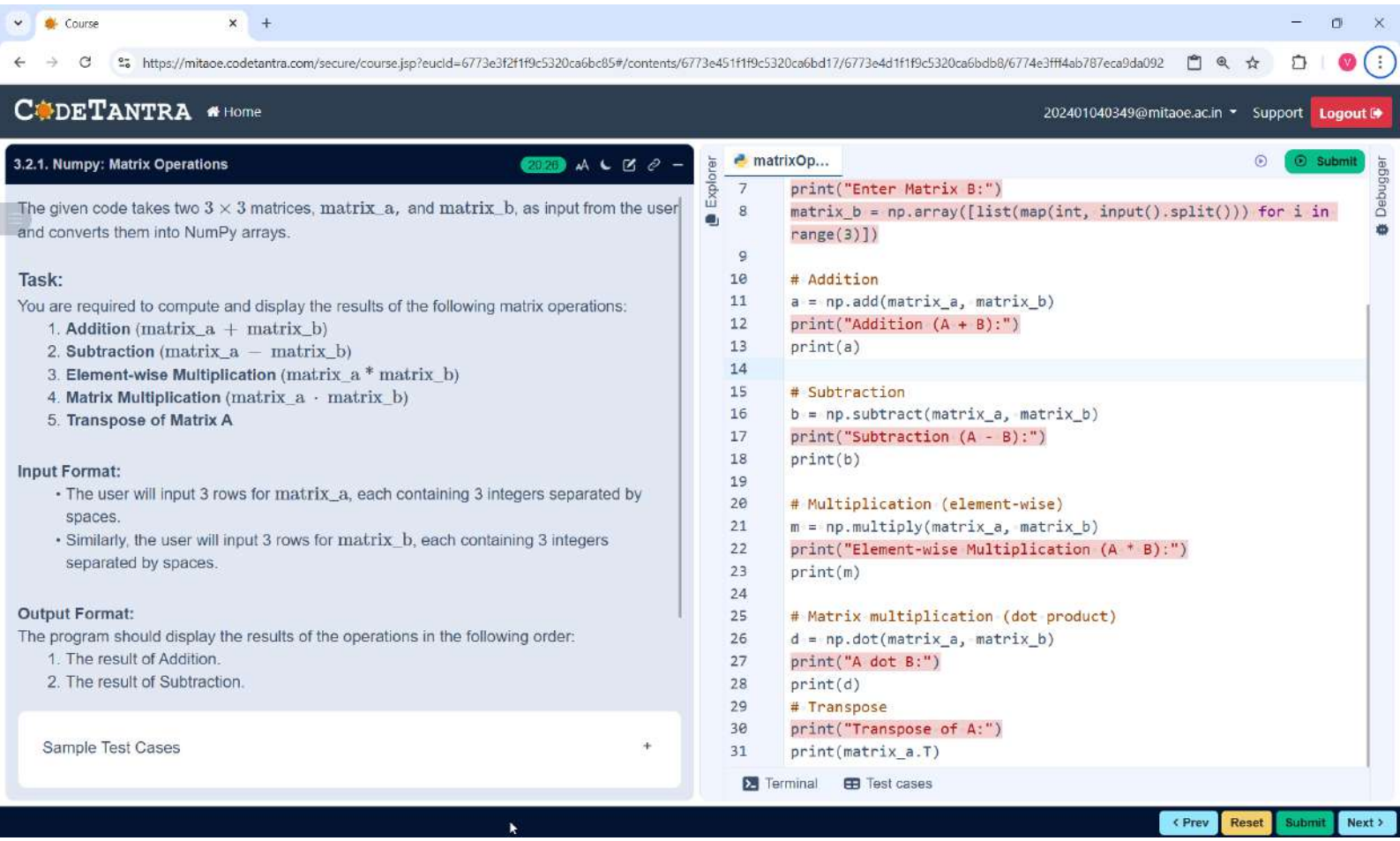
Submit

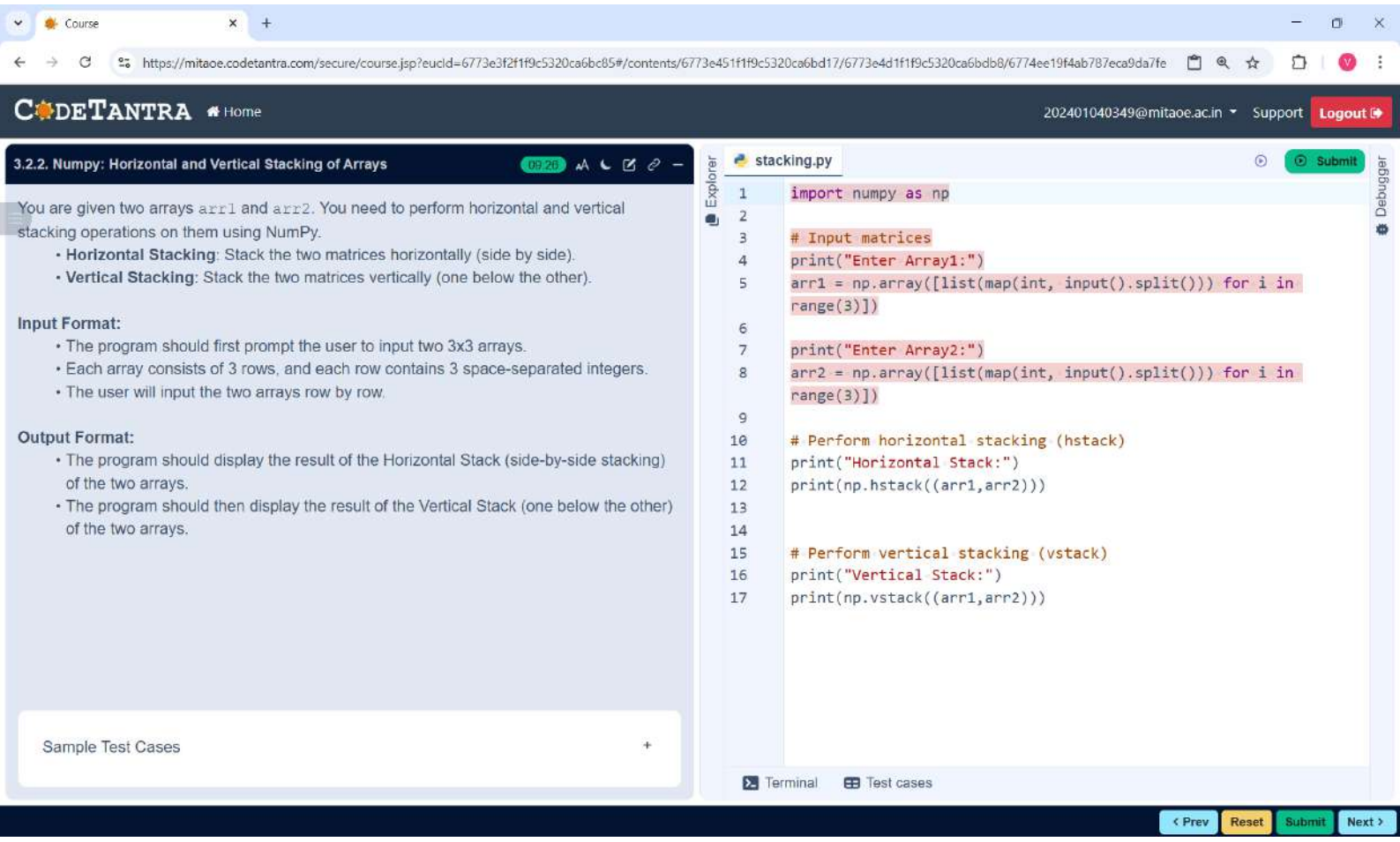
```
1  import numpy as np
2
3  # Input matrices
4  print("Enter Matrix A:")
5  matrix_a = np.array([list(map(int, input().split())) for i in
   range(3)])
6
7  print("Enter Matrix B:")
8  matrix_b = np.array([list(map(int, input().split())) for i in
   range(3)])
9
10 # Addition
11 a = np.add(matrix_a, matrix_b)
12 print("Addition (A + B):")
13 print(a)
14
15 # Subtraction
16 b = np.subtract(matrix_a, matrix_b)
17 print("Subtraction (A - B):")
18 print(b)
19
20 # Multiplication (element-wise)
21 m = np.multiply(matrix_a, matrix_b)
22 print("Element-wise Multiplication (A * B):")
23 print(m)
24
```

Terminal Test cases

Debugger

< Prev Reset Submit Next >


Course
https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e451f1f9c5320ca6bd17/6773e4d1f1f9c5320ca6bdb8/6774e3fff4ab787eca9da092
CODETANTRA
Home
202401040349@mitaoe.ac.in
Support
Logout
3.2.1. Numpy: Matrix Operations
20:26
The given code takes two $3 \times 3$ 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
7 print("Enter Matrix B:")
8 matrix_b = np.array([list(map(int, input().split())) for i in range(3)])
9
10 # Addition
11 a = np.add(matrix_a, matrix_b)
12 print("Addition (A + B):")
13 print(a)
14
15 # Subtraction
16 b = np.subtract(matrix_a, matrix_b)
17 print("Subtraction (A - B):")
18 print(b)
19
20 # Multiplication (element-wise)
21 m = np.multiply(matrix_a, matrix_b)
22 print("Element-wise Multiplication (A * B):")
23 print(m)
24
25 # Matrix multiplication (dot product)
26 d = np.dot(matrix_a, matrix_b)
27 print("A dot B:")
28 print(d)
29 # Transpose
30 print("Transpose of A:")
31 print(matrix_a.T)
Terminal
Test cases
Prev
Reset
Submit
Next


Course
https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e451f1f9c5320ca6bd17/6773e4d1f1f9c5320ca6bdb8/6774ee19f4ab787eca9da7fe
CODETANTRA
Home
202401040349@mitaoe.ac.in
Support
Logout
3.2.2. Numpy: Horizontal and Vertical Stacking of Arrays
09:26
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
1 import numpy as np
2
3 # Input matrices
4 print("Enter Array1:")
5 arr1 = np.array([list(map(int, input().split())) for i in range(3)])
6
7 print("Enter Array2:")
8 arr2 = np.array([list(map(int, input().split())) for i in range(3)])
9
10 # Perform horizontal stacking (hstack)
11 print("Horizontal Stack:")
12 print(np.hstack((arr1,arr2)))
13
14
15 # Perform vertical stacking (vstack)
16 print("Vertical Stack:")
17 print(np.vstack((arr1,arr2)))
Terminal
Test cases
< Prev
Reset
Submit
Next >

Course

https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e451f1f9c5320ca6bd17/6773e4d1f1f9c5320ca6bdb8/6774ee9df4ab787eca9da869

CODETANTRA Home

202401040349@mitaoe.ac.in Support Logout

## 3.2.3. Numpy: Custom Sequence Generation

08:45

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

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

Debugger

Terminal Test cases

< Prev Reset Submit Next >

Course

https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e451f1f9c5320ca6bd17/6773e4d1f1f9c5320ca6bdb8/6774ef2cf4ab787eca9da8c0

CODETANTRA Home 202401040349@mitaoe.ac.in Support Logout

## 3.2.4. Numpy: Arithmetic and Statistical Operations, Mathematical Opera... 20:16

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: $A_1$ `OR` $B_1$).

**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...

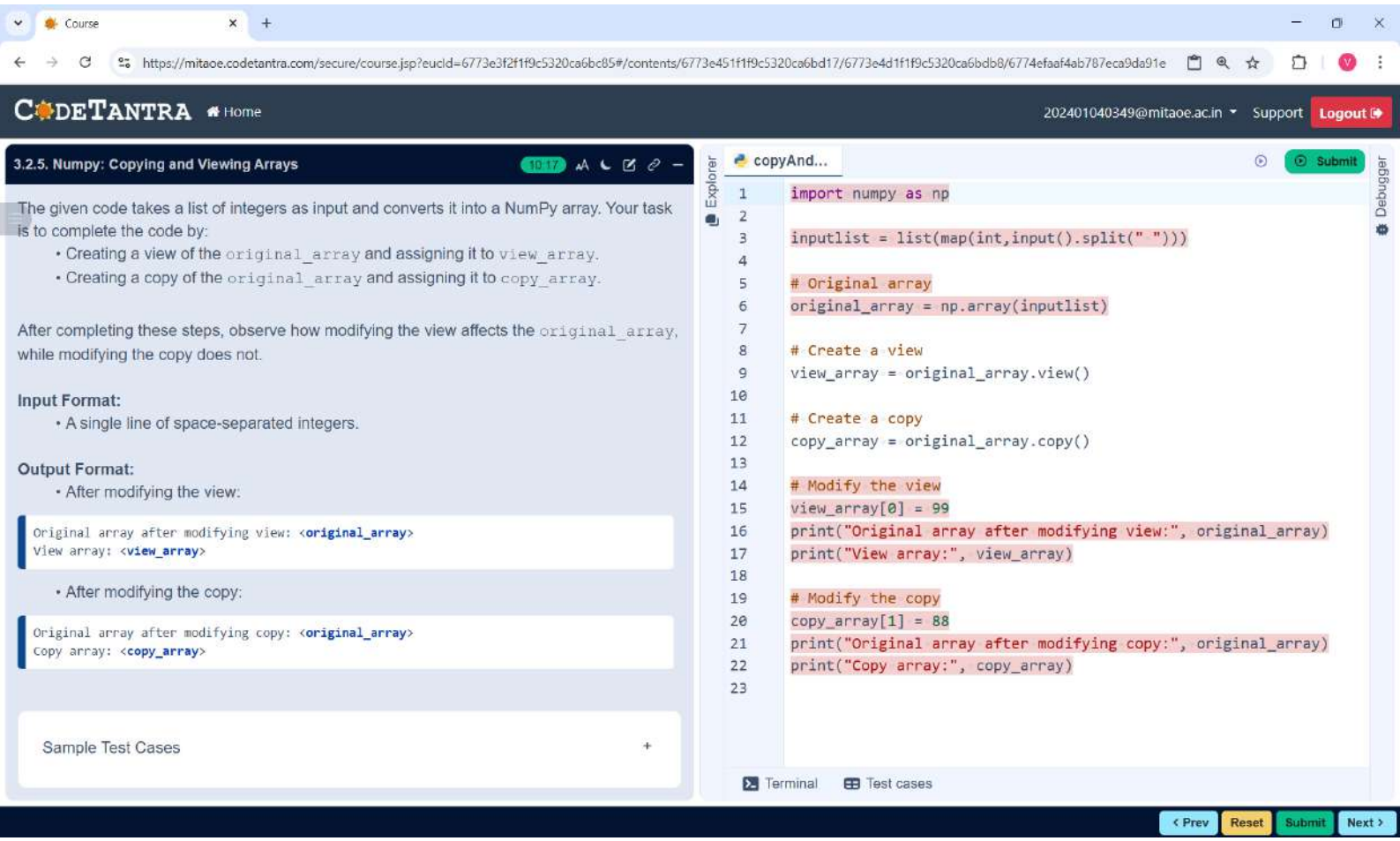
Submit

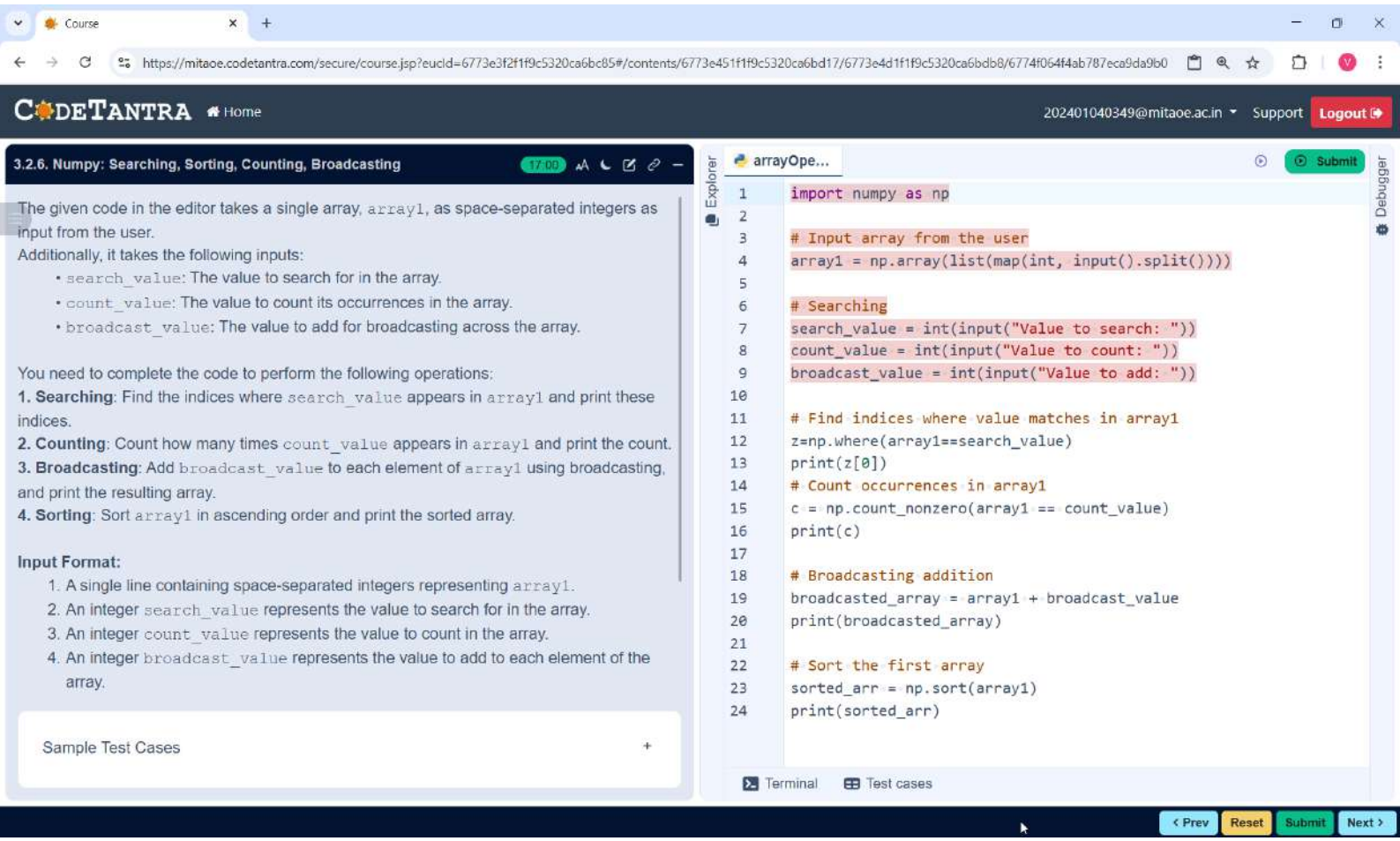
Debugger

```python
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 = np.add(A,B)
    diff_result = np.subtract(A,B)
    prod_result = np.multiply(A,B)

    # Statistical Operations
    mean_A = np.mean(A)
    median_A = np.median(A)
    std_dev_A = np.std(A)

    # Bitwise Operations
    and_result = np.bitwise_and(A,B)
    or_result = np.bitwise_or(A,B)
    xor_result = np.bitwise_xor(A,B)

    # Output results with one space between each element
    print("Element-wise Sum:", ' '.join(map(str, sum_result)))
    print("Element-wise Difference:", ' '.join(map(str,
diff_result)))
```

Terminal Test cases

< Prev Reset Submit Next >


Course
https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e451f1f9c5320ca6bd17/6773e4d1f1f9c5320ca6bdb8/6774efaaf4ab787eca9da91e
CODETANTRA
Home
202401040349@mitaoe.ac.in
Support
Logout
3.2.5. Numpy: Copying and Viewing Arrays
10:17
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
1 import numpy as np
2
3 inputlist = list(map(int,input().split(" ")))
4
5 # Original array
6 original_array = np.array(inputlist)
7
8 # Create a view
9 view_array = original_array.view()
10
11 # Create a copy
12 copy_array = original_array.copy()
13
14 # Modify the view
15 view_array[0] = 99
16 print("Original array after modifying view:", original_array)
17 print("View array:", view_array)
18
19 # Modify the copy
20 copy_array[1] = 88
21 print("Original array after modifying copy:", original_array)
22 print("Copy array:", copy_array)
23
Terminal
Test cases
< Prev
Reset
Submit
Next >


Course
https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e451f1f9c5320ca6bd17/6773e4d1f1f9c5320ca6bdb8/6774f064f4ab787eca9da9b0
CODETANTRA
Home
202401040349@mitaoe.ac.in
Support
Logout
3.2.6. Numpy: Searching, Sorting, Counting, Broadcasting
17:00
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
Explorer
arrayOpe...
Submit
Debugger
1 import numpy as np
2
3 # Input array from the user
4 array1 = np.array(list(map(int, input().split())))
5
6 # Searching
7 search_value = int(input("Value to search: "))
8 count_value = int(input("Value to count: "))
9 broadcast_value = int(input("Value to add: "))
10
11 # Find indices where value matches in array1
12 z=np.where(array1==search_value)
13 print(z[0])
14 # Count occurrences in array1
15 c = np.count_nonzero(array1 == count_value)
16 print(c)
17
18 # Broadcasting addition
19 broadcasted_array = array1 + broadcast_value
20 print(broadcasted_array)
21
22 # Sort the first array
23 sorted_arr = np.sort(array1)
24 print(sorted_arr)
Terminal
Test cases
< Prev
Reset
Submit
Next >

Course

https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e451f1f9c5320ca6bd17/6773e4d1f1f9c5320ca6bdb8/67dfb4f1d1a74a4035ef4225

CODETANTRA Home 202401040349@mitaoe.ac.in Support Logout

## 3.2.7. Student Data Analysis and Operations

41:58

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

Sample Test Cases +

Explorer

Operatio...

Submit

Debugger

```
1  import numpy as np
2
3  a = np.loadtxt("Sample.csv", delimiter=',', skiprows=1)
4
5  # 1. Print all student details
6  print("All student Details:\n", a)
7
8  # 2. print total students
9  print("Total Students:", len(a))
10
11 # 3. Print all student Roll numbers
12 print("All Student Roll Nos", a[:,0].astype(float))
13
14 # 4. Print subject 1 marks
15 print("Subject 1 Marks", a[:,1])
16
17 # 5. print minimum marks of Subject 2
18 print("Min marks in Subject 2", a[:,2].min())
19
20 # 6. print maximum marks of Subject 3
21 print("Max marks in Subject 3", a[:,3].max())
22
23 # 7. Print All subject marks
24 sagle_mark = a[:,1:]
25 print("All subject marks:",sagle_mark)
26
```

Terminal Test cases

< Prev Reset Submit Next >

Course

https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e451f1f9c5320ca6bd17/6773e4d1f1f9c5320ca6bdb8/67dfb4f1d1a74a4035ef4225

CODETANTRA Home 202401040349@mitaoe.ac.in Support Logout

## 3.2.7. Student Data Analysis and Operations

41:58

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

Sample Test Cases +

Explorer

Operatio...

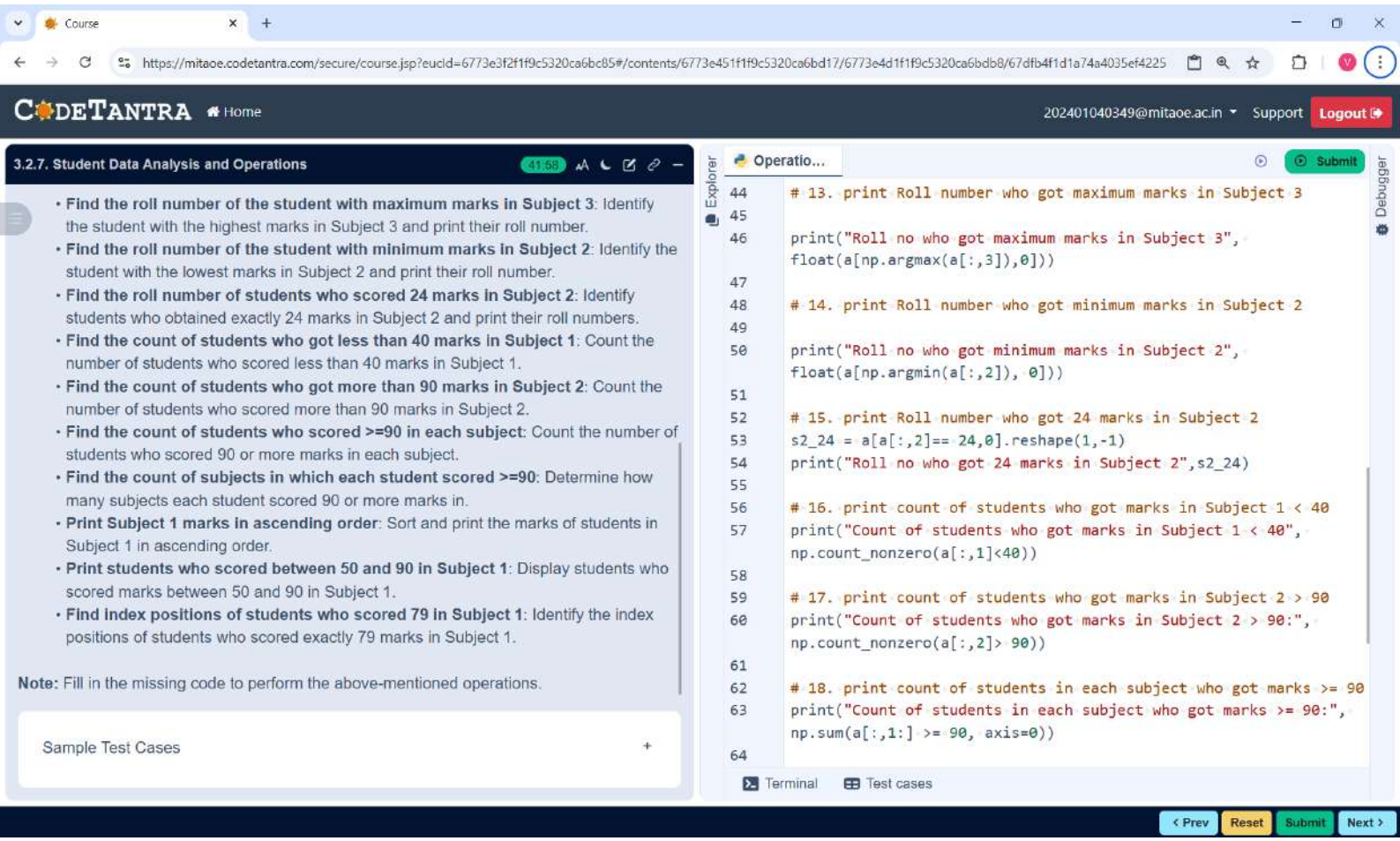
Submit

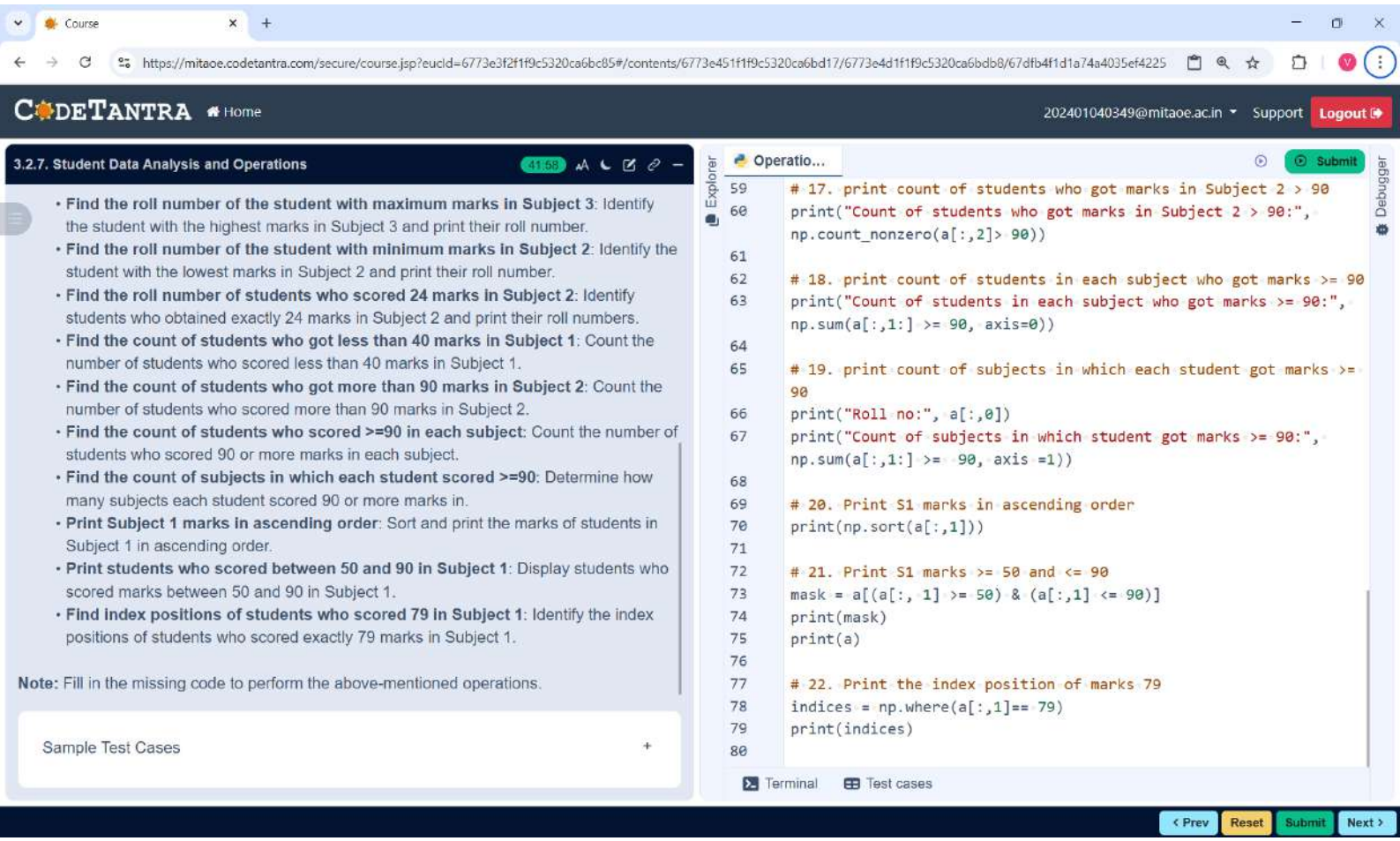
Debugger

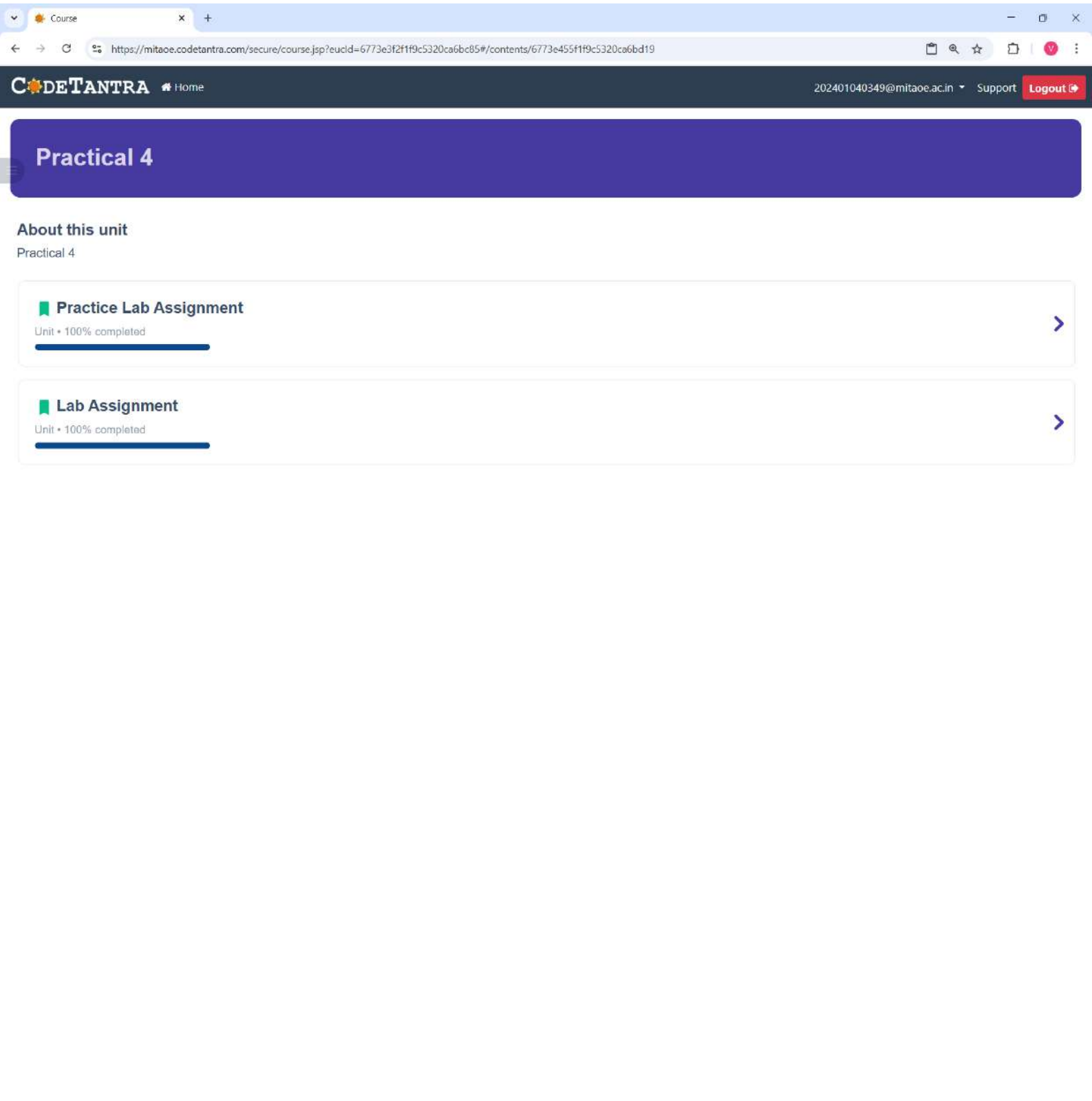
```
27  # 8. print Total marks of students
28  print("Total Marks", np.sum(a[:,1:],axis =1))
29
30  # 9. print average marks of each student
31  avg = np.mean(a[:,1:], axis =1)
32  print(np.round(avg,1))
33
34  # 10. print average marks of each subject
35  avgsub = np.mean(a[:,1:],axis =0)
36  print("Average Marks of each subject", np.round(avgsub,1))
37
38  # 11. print average marks of S1 and S2
39  print("Average Marks of S1 and S2", np.mean(a[:,1:3], axis=0) )
40
41  # 12. print average marks of S1 and S3
42  print("Average Marks of S1 and S3", np.mean(a[:,[1,3]], axis =0))
43
44  # 13. print Roll number who got maximum marks in Subject 3
45
46  print("Roll no who got maximum marks in Subject 3",
    float(a[np.argmax(a[:,3]),0]))
47
48  # 14. print Roll number who got minimum marks in Subject 2
49
50  print("Roll no who got minimum marks in Subject 2",
    float(a[np.argmin(a[:,2]), 0]))
```

Terminal Test cases

< Prev Reset Submit Next >


Course
https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e451f1f9c5320ca6bd17/6773e4d1f1f9c5320ca6bdb8/67dfb4f1d1a74a4035ef4225
CODETANTRA
Home
202401040349@mitaoe.ac.in
Support
Logout
3.2.7. Student Data Analysis and Operations
41:58
Explorer
Operatio...
Submit
Debugger
• 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.
• Find the roll number of the student with minimum marks in Subject 2: Identify the student with the lowest marks in Subject 2 and print their roll number.
• Find the roll number of students who scored 24 marks in Subject 2: Identify students who obtained exactly 24 marks in Subject 2 and print their roll numbers.
• Find the count of students who got less than 40 marks in Subject 1: Count the number of students who scored less than 40 marks in Subject 1.
• Find the count of students who got more than 90 marks in Subject 2: Count the number of students who scored more than 90 marks in Subject 2.
• Find the count of students who scored >=90 in each subject: Count the number of students who scored 90 or more marks in each subject.
• Find the count of subjects in which each student scored >=90: Determine how many subjects each student scored 90 or more marks in.
• Print Subject 1 marks in ascending order: Sort and print the marks of students in Subject 1 in ascending order.
• Print students who scored between 50 and 90 in Subject 1: Display students who scored marks between 50 and 90 in Subject 1.
• Find index positions of students who scored 79 in Subject 1: Identify the index positions of students who scored exactly 79 marks in Subject 1.
Note: Fill in the missing code to perform the above-mentioned operations.
Sample Test Cases
44 # 13. print Roll number who got maximum marks in Subject 3
45
46 print("Roll no who got maximum marks in Subject 3", float(a[np.argmax(a[:,3]),0]))
47
48 # 14. print Roll number who got minimum marks in Subject 2
49
50 print("Roll no who got minimum marks in Subject 2", float(a[np.argmin(a[:,2]), 0]))
51
52 # 15. print Roll number who got 24 marks in Subject 2
53 s2_24 = a[a[:,2]== 24,0].reshape(1,-1)
54 print("Roll no who got 24 marks in Subject 2",s2_24)
55
56 # 16. print count of students who got marks in Subject 1 < 40
57 print("Count of students who got marks in Subject 1 < 40", np.count_nonzero(a[:,1]<40))
58
59 # 17. print count of students who got marks in Subject 2 > 90
60 print("Count of students who got marks in Subject 2 > 90:", np.count_nonzero(a[:,2]> 90))
61
62 # 18. print count of students in each subject who got marks >= 90
63 print("Count of students in each subject who got marks >= 90:", np.sum(a[:,1:] >= 90, axis=0))
64
Terminal
Test cases
< Prev
Reset
Submit
Next >


Course
https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e451f1f9c5320ca6bd17/6773e4d1f1f9c5320ca6bdb8/67dfb4f1d1a74a4035ef4225
CODETANTRA
Home
202401040349@mitaoe.ac.in
Support
Logout
3.2.7. Student Data Analysis and Operations
41:58
Explorer
Operatio...
Submit
Debugger
• 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.
• Find the roll number of the student with minimum marks in Subject 2: Identify the student with the lowest marks in Subject 2 and print their roll number.
• Find the roll number of students who scored 24 marks in Subject 2: Identify students who obtained exactly 24 marks in Subject 2 and print their roll numbers.
• Find the count of students who got less than 40 marks in Subject 1: Count the number of students who scored less than 40 marks in Subject 1.
• Find the count of students who got more than 90 marks in Subject 2: Count the number of students who scored more than 90 marks in Subject 2.
• Find the count of students who scored >=90 in each subject: Count the number of students who scored 90 or more marks in each subject.
• Find the count of subjects in which each student scored >=90: Determine how many subjects each student scored 90 or more marks in.
• Print Subject 1 marks in ascending order: Sort and print the marks of students in Subject 1 in ascending order.
• Print students who scored between 50 and 90 in Subject 1: Display students who scored marks between 50 and 90 in Subject 1.
• Find index positions of students who scored 79 in Subject 1: Identify the index positions of students who scored exactly 79 marks in Subject 1.
Note: Fill in the missing code to perform the above-mentioned operations.
Sample Test Cases
59 # 17. print count of students who got marks in Subject 2 > 90
60 print("Count of students who got marks in Subject 2 > 90:", np.count_nonzero(a[:,2]> 90))
61
62 # 18. print count of students in each subject who got marks >= 90
63 print("Count of students in each subject who got marks >= 90:", np.sum(a[:,1:] >= 90, axis=0))
64
65 # 19. print count of subjects in which each student got marks >= 90
66 print("Roll no:", a[:,0])
67 print("Count of subjects in which student got marks >= 90:", np.sum(a[:,1:] >= 90, axis =1))
68
69 # 20. Print S1 marks in ascending order
70 print(np.sort(a[:,1]))
71
72 # 21. Print S1 marks >= 50 and <= 90
73 mask = a[(a[:, 1] >= 50) & (a[:,1] <= 90)]
74 print(mask)
75 print(a)
76
77 # 22. Print the index position of marks 79
78 indices = np.where(a[:,1]== 79)
79 print(indices)
80
Terminal
Test cases
< Prev
Reset
Submit
Next >


Course
https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e455f1f9c5320ca6bd19
CODETANTRA
Home
202401040349@mitaoe.ac.in
Support
Logout
Practical 4
About this unit
Practical 4
Practice Lab Assignment
Unit • 100% completed
Lab Assignment
Unit • 100% completed

Course

https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e455f1f9c5320ca6bd19/6773e4d5f1f9c5320ca6bdc2/667a7e0428220463f6497a3c

CODETANTRA Home 202401040349@mitaoe.ac.in Support Logout

## 4.1.1. Pandas - series creation and manipulation

33.54

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 +

Explorer

seriesMa...

Submit

Debugger

```
1  import pandas as pd
2
3  # Take inputs from the user to create a list of numbers
4  numbers = list(map(int, input().split()))
5
6  # Create a Pandas series from the list of numbers
7  ser = pd.Series(numbers)
8
9  # Grouping by even and odd numbers and calculating the mean
10 grouped =ser.groupby(ser%2==0).mean()
11
12 # Display the mean of even and odd numbers with labels
13 grouped.index = ['Even' if is_even else 'Odd' for is_even in
   grouped.index]
14 print("Mean of even and odd numbers:")
15 print(grouped)
16
```

Terminal Test cases

< Prev Reset Submit Next >

Course

https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e455f1f9c5320ca6bd19/6773e4d5f1f9c5320ca6bdc2/667aa30728220463f649bb62

CODETANTRA Home 202401040349@mitaoe.ac.in Support Logout

## 4.1.2. Dictionary to dataframe

29.31

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.

Sample Test Cases +

Explorer

dataframe...

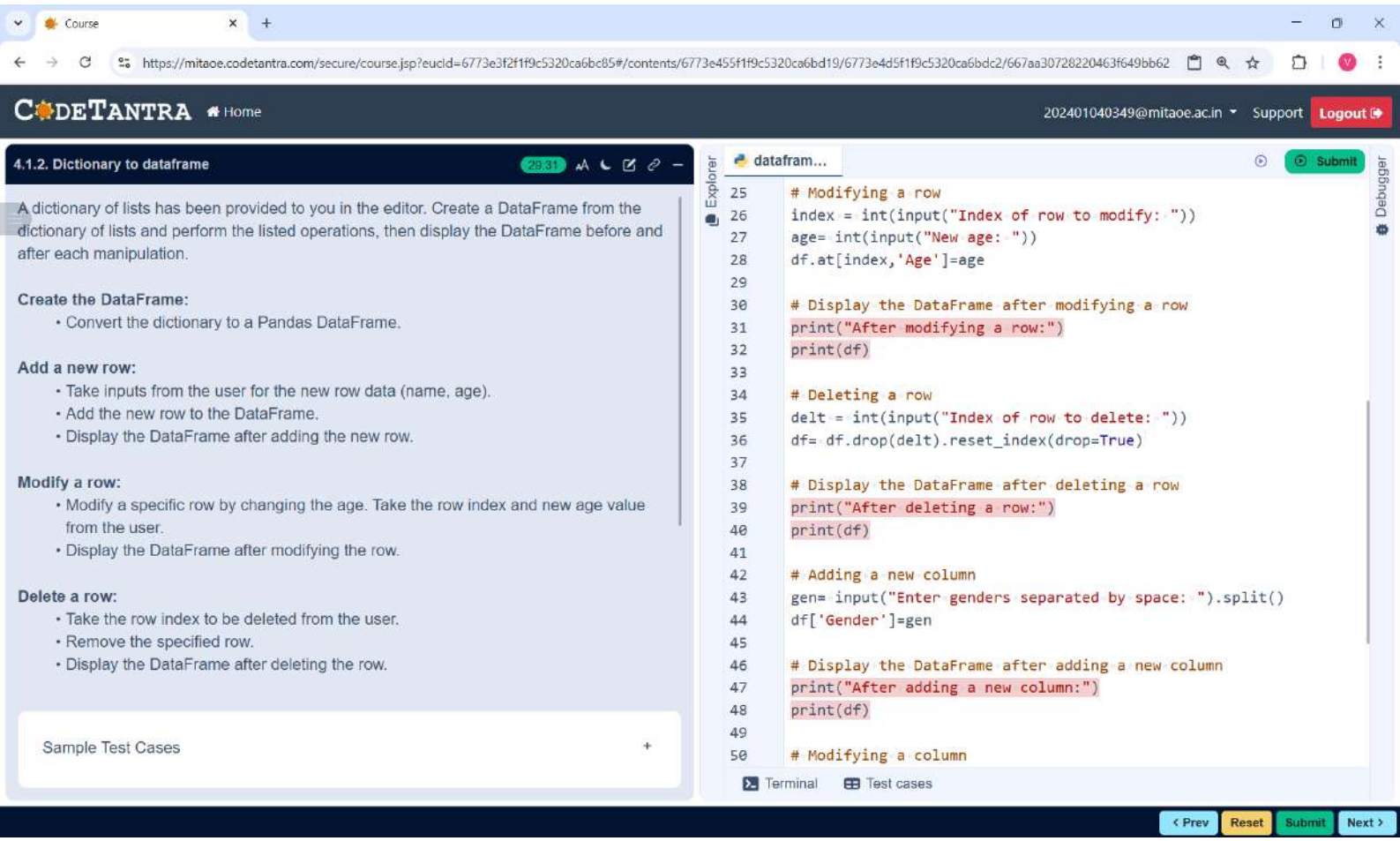
Submit

Debugger

```
1  import pandas as pd
2
3  # Provided dictionary of lists
4  data = {
5      'Name': ['Alice', 'Bob', 'Charlie'],
6      'Age': [25, 30, 35],
7  }
8
9  # Convert the dictionary to a DataFrame
10 df = pd.DataFrame(data)
11
12 # Display the original DataFrame
13 print("Original DataFrame:")
14 print(df)
15
16 # Adding a new row
17 Name = input("New name: ")
18 Age = int(input("New age: "))
19 new = {'Name':Name,'Age':Age}
20 df=df.append(new,ignore_index=True)
21
22 # Display the DataFrame after adding a new row
23 print("After adding a row:\n",df)
24
25 # Modifying a row
26 index = int(input("Index of row to modify: "))
```

Terminal Test cases

< Prev Reset Submit Next >


Course
https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e455f1f9c5320ca6bd19/6773e4d5f1f9c5320ca6bdc2/667aa30728220463f649bb62
CODETANTRA
Home
202401040349@mitaoe.ac.in
Support
Logout
4.1.2. Dictionary to dataframe
29.31
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.
Sample Test Cases
Explorer
datafram...
Submit
Debugger
25 # Modifying a row
26 index = int(input("Index of row to modify: "))
27 age= int(input("New age: "))
28 df.at[index,'Age']=age
29
30 # Display the DataFrame after modifying a row
31 print("After modifying a row:")
32 print(df)
33
34 # Deleting a row
35 delt = int(input("Index of row to delete: "))
36 df= df.drop(delt).reset_index(drop=True)
37
38 # Display the DataFrame after deleting a row
39 print("After deleting a row:")
40 print(df)
41
42 # Adding a new column
43 gen= input("Enter genders separated by space: ").split()
44 df['Gender']=gen
45
46 # Display the DataFrame after adding a new column
47 print("After adding a new column:")
48 print(df)
49
50 # Modifying a column
Terminal
Test cases
< Prev
Reset
Submit
Next >

Course

https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e455f1f9c5320ca6bd19/6773e4d5f1f9c5320ca6bdc2/667aa30728220463f649bb62

CODETANTRA Home 202401040349@mitaoe.ac.in Support Logout

## 4.1.2. Dictionary to dataframe

29.31

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.

Sample Test Cases +

Explorer

dataframe...

Submit

Debugger

```
38  # Display the DataFrame after deleting a row
39  print("After deleting a row:")
40  print(df)
41
42  # Adding a new column
43  gen= input("Enter genders separated by space: ").split()
44  df['Gender']=gen
45
46  # Display the DataFrame after adding a new column
47  print("After adding a new column:")
48  print(df)
49
50  # Modifying a column
51  df['Name']=df['Name'].str.upper()
52
53  # Display the DataFrame after modifying a column
54  print("After modifying a column:")
55  print(df)
56
57  # Deleting a column
58  df=df.drop(columns=['Age'])
59
60  # Display the DataFrame after deleting a column
61  print("After deleting a column:")
62  print(df)
63
```

Terminal Test cases

< Prev Reset Submit Next >

Course

https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e455f1f9c5320ca6bd19/6773e4d5f1f9c5320ca6bdc2/667cf60f2365ba61af5a2ce7

CODETANTRA Home 202401040349@mitaoe.ac.in Support Logout

## 4.1.3. Student Information

11:35

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 +

Explorer

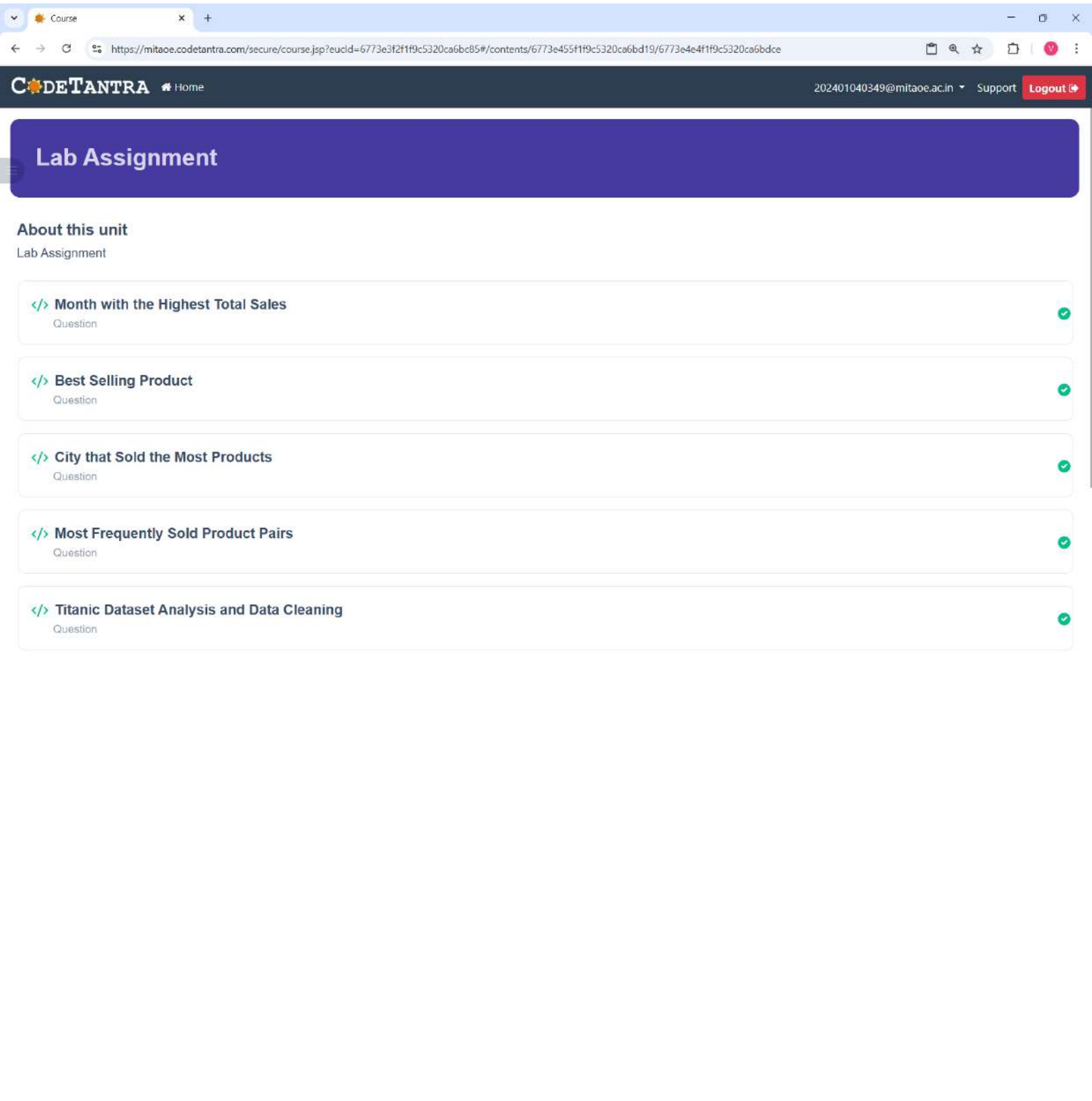
studentin... studentdat... Submit

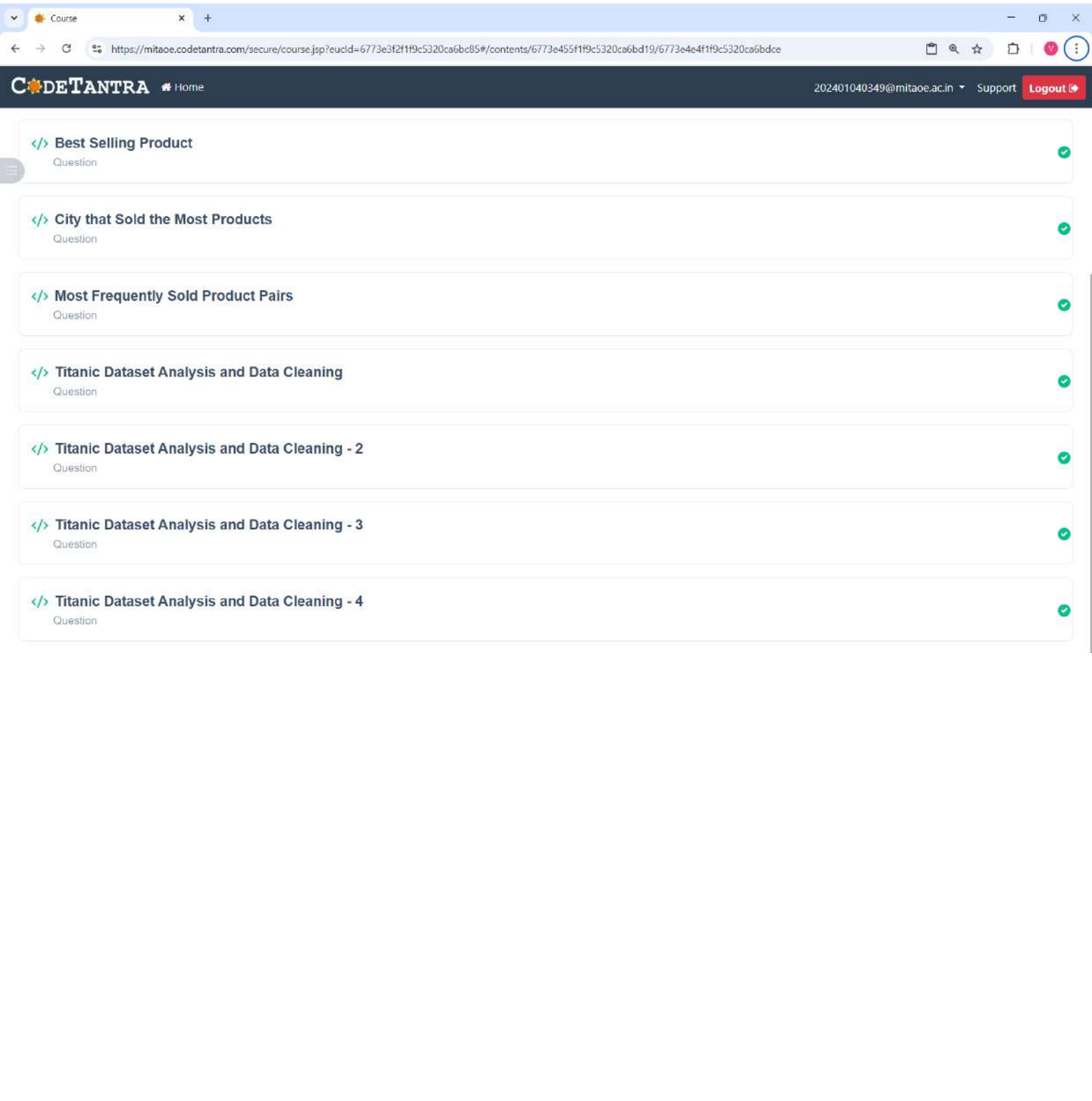
```
1  import pandas as pd
2
3  # Read the text file into a DataFrame
4  file = input()
5  data = pd.read_csv(file, sep="\s+", header=None, names=["Name",
   "Age", "Grade"])
6  print("First five rows:")
7  print(data.head())
8  avg=round(data['Age'].mean(),2)
9  print("Average age:",avg)
10
11
12 # write your code here..
13
14 filter=data[data['Grade'] <= 'B']
15 print("Students with a grade up to B")
16 print(filter)
17
```

Debugger

Terminal Test cases

< Prev Reset Submit Next >


Course
https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e455f1f9c5320ca6bd19/6773e4e4f1f9c5320ca6bdce
CODETANTRA
Home
202401040349@mitaoe.ac.in
Support
Logout
Lab Assignment
About this unit
Lab Assignment
Month with the Highest Total Sales
Question
Best Selling Product
Question
City that Sold the Most Products
Question
Most Frequently Sold Product Pairs
Question
Titanic Dataset Analysis and Data Cleaning
Question


Course
https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e455f1f9c5320ca6bd19/6773e4e4f1f9c5320ca6bdce
CODETANTRA
Home
202401040349@mitaoe.ac.in
Support
Logout
Best Selling Product
Question
City that Sold the Most Products
Question
Most Frequently Sold Product Pairs
Question
Titanic Dataset Analysis and Data Cleaning
Question
Titanic Dataset Analysis and Data Cleaning - 2
Question
Titanic Dataset Analysis and Data Cleaning - 3
Question
Titanic Dataset Analysis and Data Cleaning - 4
Question

Course

https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e455f1f9c5320ca6bd19/6773e4e4f1f9c5320ca6bdce/67860994a50fe56b8dcf8747

CODETANTRA Home 202401040349@mitaoe.ac.in Support Logout

## 4.2.1. Month with the Highest Total Sales

18:06

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 +

Explorer

monthFor... sales_dat...

Submit

Debugger

```
1  import pandas as pd
2
3  # Prompt the user for the file name
4  file_name = input()
5
6  # Load the data
7  df = pd.read_csv(file_name)
8
9  df['Total_Sales']=df['Quantity']*df['Price']
10 df['Date']=pd.to_datetime(df['Date'])
11 df['Month']=df['Date'].dt.to_period('M')
12 monthly_sales=df.groupby('Month')['Total_Sales'].sum()
13
14 # Find the month with the highest total sales
15 best_month = monthly_sales.idxmax()
16 highest_sales = monthly_sales.max()
17 print(f"Best month: {best_month}")
18
19 #print(f"Best month: {best_month}")
20 print(f"Total sales: ${highest_sales:.2f}")
21
```

Terminal Test cases

Show hidden icons

Reset Submit Next >

Course

https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e455f1f9c5320ca6bd19/6773e4e4f1f9c5320ca6bdce/67861370a50fe56b8dcf9828

CODETANTRA Home 202401040349@mitaoe.ac.in Support Logout

## 4.2.2. Best Selling Product

42:18

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`.
- 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

Sample Test Cases +

Explorer

monthFor... sales_dat... Submit

```python
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().reset_index()

# Find the product with the highest total quantity sold
best_product =
product_sales.loc[product_sales['Quantity'].idxmax()]
highest_quantity = product_sales.max()
print(f"Best selling product: {best_product['Product']}")
print(f"Total quantity sold: {best_product['Quantity']}")

"""
# Display the result
print(f"Best selling product: {best_product}")
print(f"Total quantity sold: {highest_quantity}")

"""
```

Debugger

Terminal Test cases

< Prev Reset Submit Next >

Course

https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e455f1f9c5320ca6bd19/6773e4e4f1f9c5320ca6bdce/67861553a50fe56b8dcf9a6e

CODETANTRA Home 202401040349@mitaoe.ac.in Support Logout

## 4.2.3. City that Sold the Most Products

04:23

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

Sample Test Cases +

Explorer

monthFor... sales_dat...

Submit

```python
import pandas as pd

# Prompt the user for the file name
file_name = input()

# Load the data
df = pd.read_csv(file_name)
city_wise_sell=df.groupby('City')['Quantity'].sum().reset_index()

best_city=city_wise_sell.loc[city_wise_sell['Quantity'].idxmax()]

"""
# Display the result
print(f"City sold the most products: {best_city}")
"""
print(f"City sold the most products: {best_city['City']}")
```

Debugger

Terminal Test cases

< Prev Reset Submit Next >

Course

https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e455f1f9c5320ca6bd19/6773e4e4f1f9c5320ca6bdce/6788eea89ff79b6ea9e3ece1

CODETANTRA Home 202401040349@mitaoe.ac.in Support Logout

## 4.2.4. Most Frequently Sold Product Pairs

06:36

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:**

- **2025-01-01**: Product A, Product B

Sample Test Cases +

Explorer

frequentl... sales_dat...

Submit

Debugger

```
1  import pandas as pd
2  from itertools import combinations
3  from collections import Counter
4
5  # Prompt user to input the file name
6  file_name = input()
7
8  # Read data from the specified CSV file
9  df = pd.read_csv(file_name)
10
11 # write the code
12 grouped=df.groupby('Date')['Product'].apply(list)
13
14 #create list
15 product_combinations=[]
16 for products in grouped:
17     product_combinations.extend(combinations(sorted(set(products)),2)
   )
18
19 combinations_count= Counter(product_combinations)
20 max_count=combinations_count.most_common(1)[0][1]
21
22 # Output the most frequent product pairs
23 for combo, count in combinations_count.items():
24     if count==max_count:
```

Terminal Test cases

< Prev Reset Submit Next >

Course

https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e455f1f9c5320ca6bd19/6773e4e4f1f9c5320ca6bdce/67e2b9309025661df3aa574f

CODETANTRA Home 202401040349@mitaoe.ac.in Support Logout

## 4.2.5. Titanic Dataset Analysis and Data Cleaning

11:21

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,

| Passenger | Survived | Pclass | Name | Sex | Age | SibSp | Parch | Ticket | Fare | Cabin | Embarked |
|---|---|---|---|---|---|---|---|---|---|---|---|

Sample Test Cases +

Explorer

titanicDat...

Submit

Debugger

```python
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())

# 2. Display the last 5 rows of the dataset
print(data.tail())

# 3. Get the shape of the dataset
print(data.shape)

# 4. Get a summary of the dataset (info)
print(data.info())

# 5. Get basic statistics of the dataset
print(data.describe())

# 6. Check for missing values
print(data.isnull().sum())

# 7. Fill missing values in the 'Age' column with the median age
data['Age'].fillna(data['Age'].median(),inplace=True)
```

Terminal Test cases

< Prev Reset Submit Next >

Course

https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e455f1f9c5320ca6bd19/6773e4e4f1f9c5320ca6bdce/67e2b9309025661df3aa574f

CODETANTRA Home 202401040349@mitaoe.ac.in Support Logout

## 4.2.5. Titanic Dataset Analysis and Data Cleaning

11:21

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,

| Passenger | Survived | Pclass | Name | Sex | Age | SibSp | Parch | Ticket | Fare | Cabin | Embarked |
|---|---|---|---|---|---|---|---|---|---|---|---|

Sample Test Cases +

Explorer

titanicDat...

Submit

Debugger

```
12
13  # 3. Get the shape of the dataset
14  print(data.shape)
15
16  # 4. Get a summary of the dataset (info)
17  print(data.info())
18
19  # 5. Get basic statistics of the dataset
20  print(data.describe())
21
22  # 6. Check for missing values
23  print(data.isnull().sum())
24
25  # 7. Fill missing values in the 'Age' column with the median age
26  data['Age'].fillna(data['Age'].median(),inplace=True)
27
28  # 8. Fill missing values in the 'Embarked' column with the mode
29  data['Embarked'].fillna(data['Embarked'].mode()[0],inplace=True)
30
31  # 9. Drop the 'Cabin' column due to many missing values
32  data.drop('Cabin',axis=1, inplace=True)
33
34  # 10. Create a new column 'FamilySize' by adding 'SibSp' and
    'Parch'
35  data['FamilySize']=data['SibSp']+ data['Parch']
36
```

Terminal Test cases

< Prev Reset Submit Next >

Course

https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e455f1f9c5320ca6bd19/6773e4e4f1f9c5320ca6bdce/67e378a17028471d94dad0a6

CODETANTRA Home 202401040349@mitaoe.ac.in Support Logout

## 4.2.6. Titanic Dataset Analysis and Data Cleaning - 2

18:50

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,

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

Sample Test Cases +

Explorer

titanicDat...

Submit

Debugger

```python
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']=np.where(data['FamilySize']>0,0,1)

# 2. Convert 'Sex' to numeric (male: 0, female: 1)
data['Sex']=data['Sex'].map({'male':0,'female':1})

# 3. One-hot encode the 'Embarked' column
data=pd.get_dummies(data,columns=['Embarked'],drop_first=True)

# 4. Get the mean age of passengers
print(data['Age'].mean())

# 5. Get the median fare of passengers
print(data['Fare'].median())

# 6. Get the number of passengers by class
print(data['Pclass'].value_counts())


```

Terminal Test cases

< Prev Reset Submit Next >

Course

https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e455f1f9c5320ca6bd19/6773e4e4f1f9c5320ca6bdce/67e378a17028471d94dad0a6

CODETANTRA Home

202401040349@mitaoe.ac.in Support Logout

## 4.2.6. Titanic Dataset Analysis and Data Cleaning - 2

18:50

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,

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

Sample Test Cases +

Explorer

titanicDat...

Submit

Debugger

```
17  # 4. Get the mean age of passengers
18  print(data['Age'].mean())
19
20  # 5. Get the median fare of passengers
21  print(data['Fare'].median())
22
23  # 6. Get the number of passengers by class
24  print(data['Pclass'].value_counts())
25
26
27  # 7. Get the number of passengers by gender
28  print(data['Sex'].value_counts())
29
30
31  # 8. Get the number of passengers by survival status
32  print(data['Survived'].value_counts())
33
34
35  # 9. Calculate the survival rate
36  print(data['Survived'].mean())
37
38
39  # 10. Calculate the survival rate by gender
40  print(data.groupby('Sex')['Survived'].mean())
41
42
```

Terminal Test cases

< Prev Reset Submit Next >

Course

https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e455f1f9c5320ca6bd19/6773e4e4f1f9c5320ca6bdce/67e3805d7028471d94dae951

CODETANTRA Home 202401040349@mitaoe.ac.in Support Logout

## 4.2.7. Titanic Dataset Analysis and Data Cleaning - 3

49:11

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,

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

Sample Test Cases +

Explorer

titanicDat...

Submit

Debugger

```python
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
print(data.groupby('Pclass')['Survived'].mean())

# 2. Calculate the survival rate by embarked location
print(data.groupby('Embarked_S')['Survived'].mean())

# 3. Calculate the survival rate by family size
print(data.groupby('FamilySize')['Survived'].mean())

# 4. Calculate the survival rate by being alone
print(data.groupby('IsAlone')['Survived'].mean())

# 5. Get the average fare by class
print(data.groupby('Pclass')['Fare'].mean())

# 6. Get the average age by class
```

Terminal Test cases

< Prev Reset Submit Next >

Course

https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e455f1f9c5320ca6bd19/6773e4e4f1f9c5320ca6bdce/67e3805d7028471d94dae951

CODETANTRA Home 202401040349@mitaoe.ac.in Support Logout

## 4.2.7. Titanic Dataset Analysis and Data Cleaning - 3

49:11

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,

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

Sample Test Cases +

Explorer

titanicDat...

Submit

Debugger

```
13  # 2. Calculate the survival rate by embarked location
14  print(data.groupby('Embarked_S')['Survived'].mean())
15
16  # 3. Calculate the survival rate by family size
17  print(data.groupby('FamilySize')['Survived'].mean())
18
19  # 4. Calculate the survival rate by being alone
20  print(data.groupby('IsAlone')['Survived'].mean())
21
22  # 5. Get the average fare by class
23  print(data.groupby('Pclass')['Fare'].mean())
24
25  # 6. Get the average age by class
26  print(data.groupby('Pclass')['Age'].mean())
27
28  # 7. Get the average age by survival status
29  print(data.groupby('Survived')['Age'].mean())
30
31  # 8. Get the average fare by survival status
32  print(data.groupby('Survived')['Fare'].mean())
33
34  # 9. Get the number of survivors by class
35  print(data[data['Survived']==1]['Pclass'].value_counts())
36
37  # 10. Get the number of non-survivors by class
38  print(data[data['Survived']==0]['Pclass'].value_counts())
```

Terminal Test cases

< Prev Reset Submit Next >

Course

https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e455f1f9c5320ca6bd19/6773e4e4f1f9c5320ca6bdce/67e387fc7028471d94db0238

CODETANTRA Home 202401040349@mitaoe.ac.in Support Logout

## 4.2.8. Titanic Dataset Analysis and Data Cleaning - 4

08:26

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 +

Explorer

titanicDat...

Submit

Debugger

```
1  import pandas as pd
2  import numpy as np
3
4  # Load the Titanic dataset
5  data = pd.read_csv('Titanic-Dataset.csv')
6  data = pd.get_dummies(data, columns=['Embarked'],
   drop_first=True)
7
8
9  # 1. Get the number of survivors by gender
10 print(data[data['Survived']==1]['Sex'].value_counts())
11
12 # 2. Get the number of non-survivors by gender
13 print(data[data['Survived']==0]['Sex'].value_counts())
14
15 # 3. Get the number of survivors by embarked location
16 print(data[data['Survived']==1]['Embarked_S'].value_counts())
17
18 # 4. Get the number of non-survivors by embarked location
19 print(data[data['Survived']==0]['Embarked_S'].value_counts())
20
21 # 5. Calculate the percentage of children (Age < 18) who survived
22 children = data[data['Age']<18]
23 print(children['Survived'].mean())
24
25 # 6. Calculate the percentage of adults (Age >= 18) who survived
```

Terminal Test cases

< Prev Reset Submit Next >

Course

https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e455f1f9c5320ca6bd19/6773e4e4f1f9c5320ca6bdce/67e387fc7028471d94db0238

CODETANTRA Home 202401040349@mitaoe.ac.in Support Logout

## 4.2.8. Titanic Dataset Analysis and Data Cleaning - 4

08:26

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 |
|---|---|---|---|---|---|---|---|---|---|---|---|

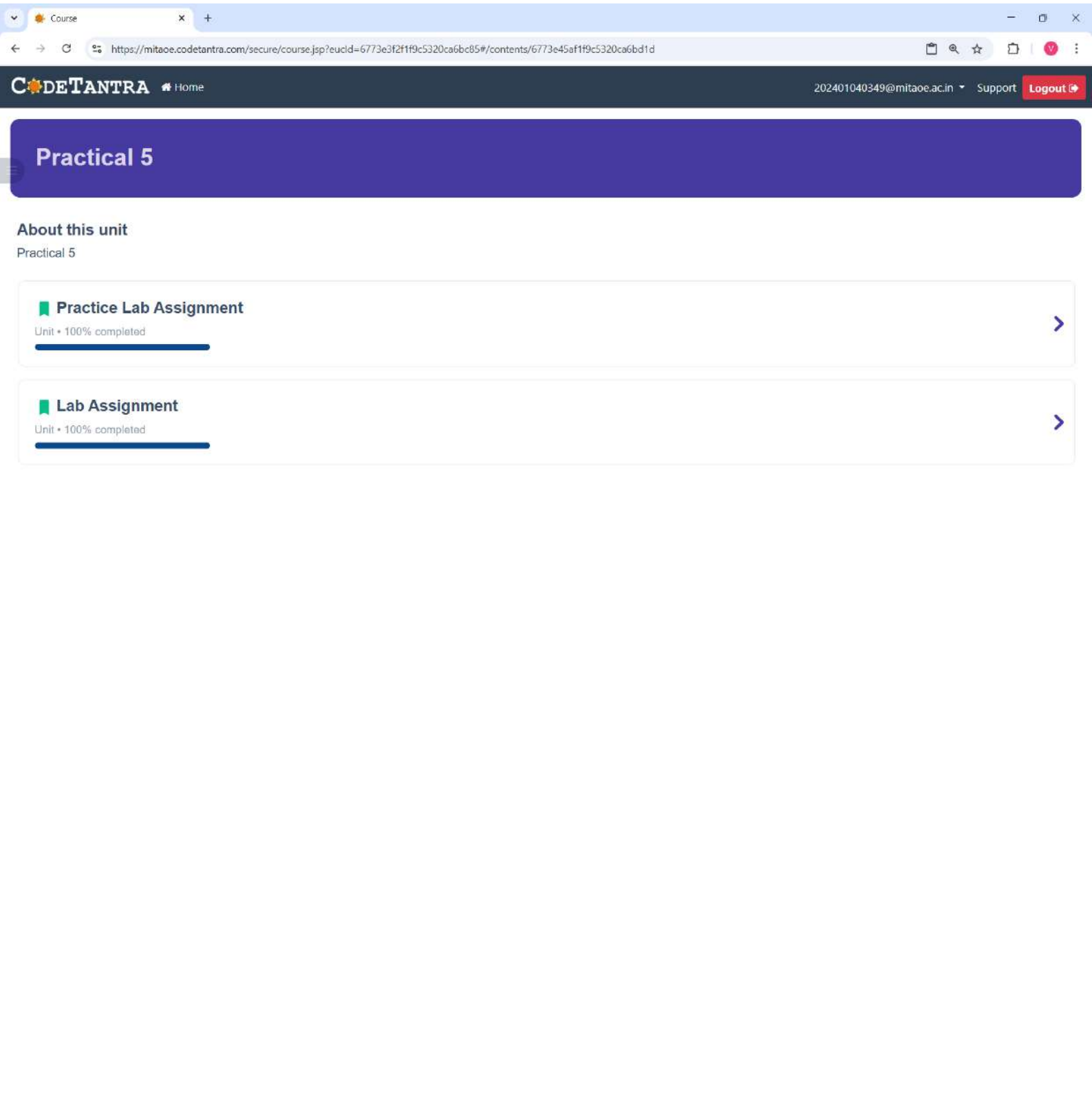
Sample Test Cases +

Explorer titanicDat... Submit Debugger

```
14
15  # 3. Get the number of survivors by embarked location
16  print(data[data['Survived']==1]['Embarked_S'].value_counts())
17
18  # 4. Get the number of non-survivors by embarked location
19  print(data[data['Survived']==0]['Embarked_S'].value_counts())
20
21  # 5. Calculate the percentage of children (Age < 18) who survived
22  children = data[data['Age']<18]
23  print(children['Survived'].mean())
24
25  # 6. Calculate the percentage of adults (Age >= 18) who survived
26  adults = data[data['Age'] >= 18]
27  print(adults['Survived'].mean())
28
29  # 7. Get the median age of survivors
30  print(data[data['Survived']==1]['Age'].median())
31
32  # 8. Get the median age of non-survivors
33  print(data[data['Survived']==0]['Age'].median())
34
35  # 9. Get the median fare of survivors
36  print(data[data['Survived']==1]['Fare'].median())
37
38  # 10. Get the median fare of non-survivors
39  print(data[data['Survived']==0]['Fare'].median())
```

Terminal Test cases

< Prev Reset Submit Next >


Course
https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e45af1f9c5320ca6bd1d
CODETANTRA
Home
202401040349@mitaoe.ac.in
Support
Logout
Practical 5
About this unit
Practical 5
Practice Lab Assignment
Unit • 100% completed
Lab Assignment
Unit • 100% completed

Course

https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e45af1f9c5320ca6bd1d/6773e4e9f1f9c5320ca6bdd2/675a8d2c09f9fa022faa9dc0

CODETANTRA Home

202401040349@mitaoe.ac.in Support Logout

## 5.1.1. Stacked Plot

10:02

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 +

Explorer

stackedpl...

Submit

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 }
11 #Create a dataframe from the data
12 df = pd.DataFrame(data)
13
14 #Creating stacked area plot
15 plt.stackplot(df['Month'],df['City_A_Temperature'],df['City_B_Tem
   perature'],df['City_C_Temperature'])
16
17 #Adding labels and titles
18 plt.xlabel('Month')
19 plt.ylabel('Temperature')
20 plt.title('Temperature Variation')
```

Terminal Test cases

< Prev Reset Submit Next >

Course

https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e45af1f9c5320ca6bd1d/6773e4e9f1f9c5320ca6bdd2/675a8d2c09f9fa022faa9dc0

CODETANTRA Home 202401040349@mitaoe.ac.in Support Logout

## 5.1.1. Stacked Plot

10:02

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 +

Explorer

stackedpl...

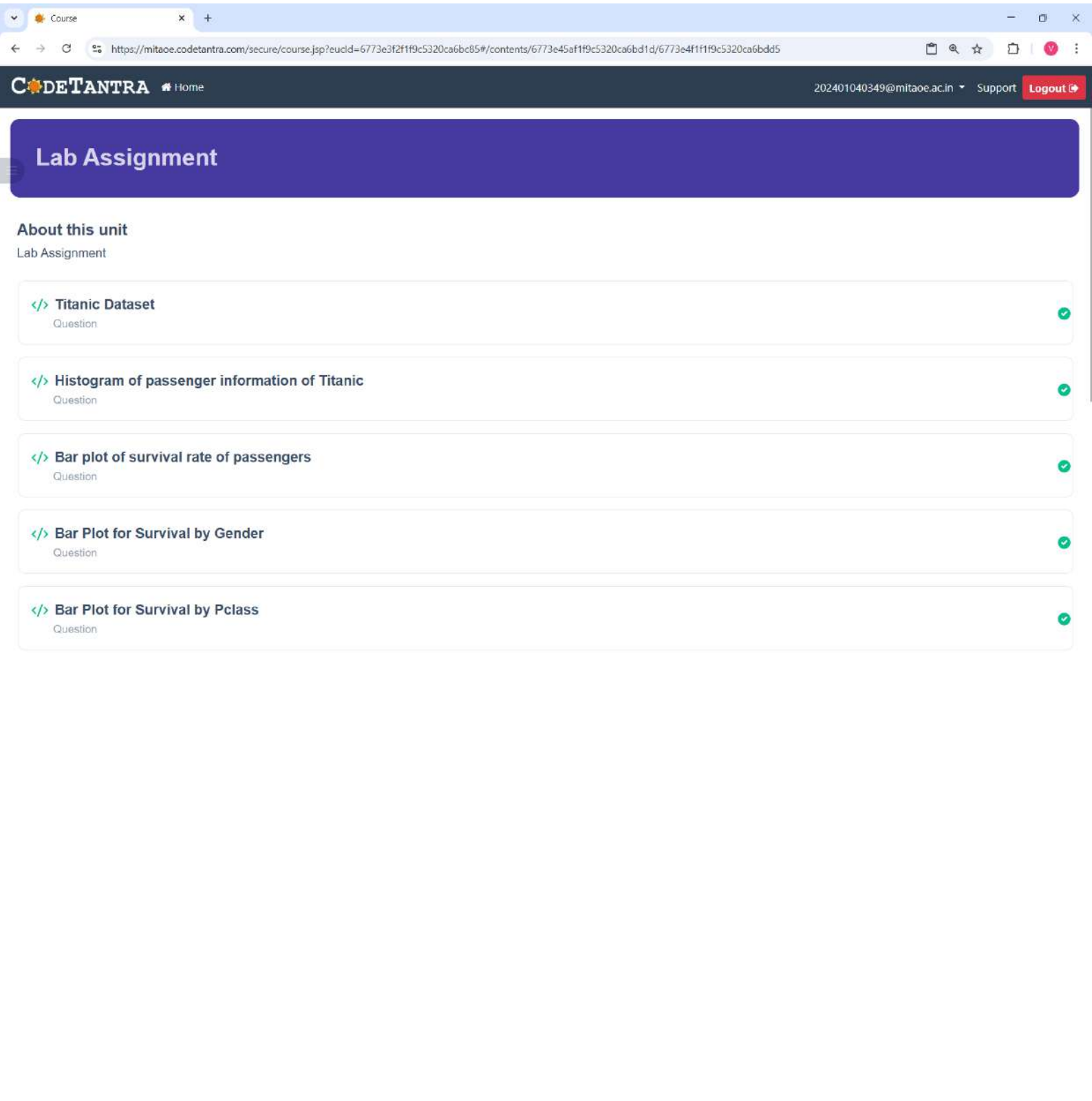
Submit

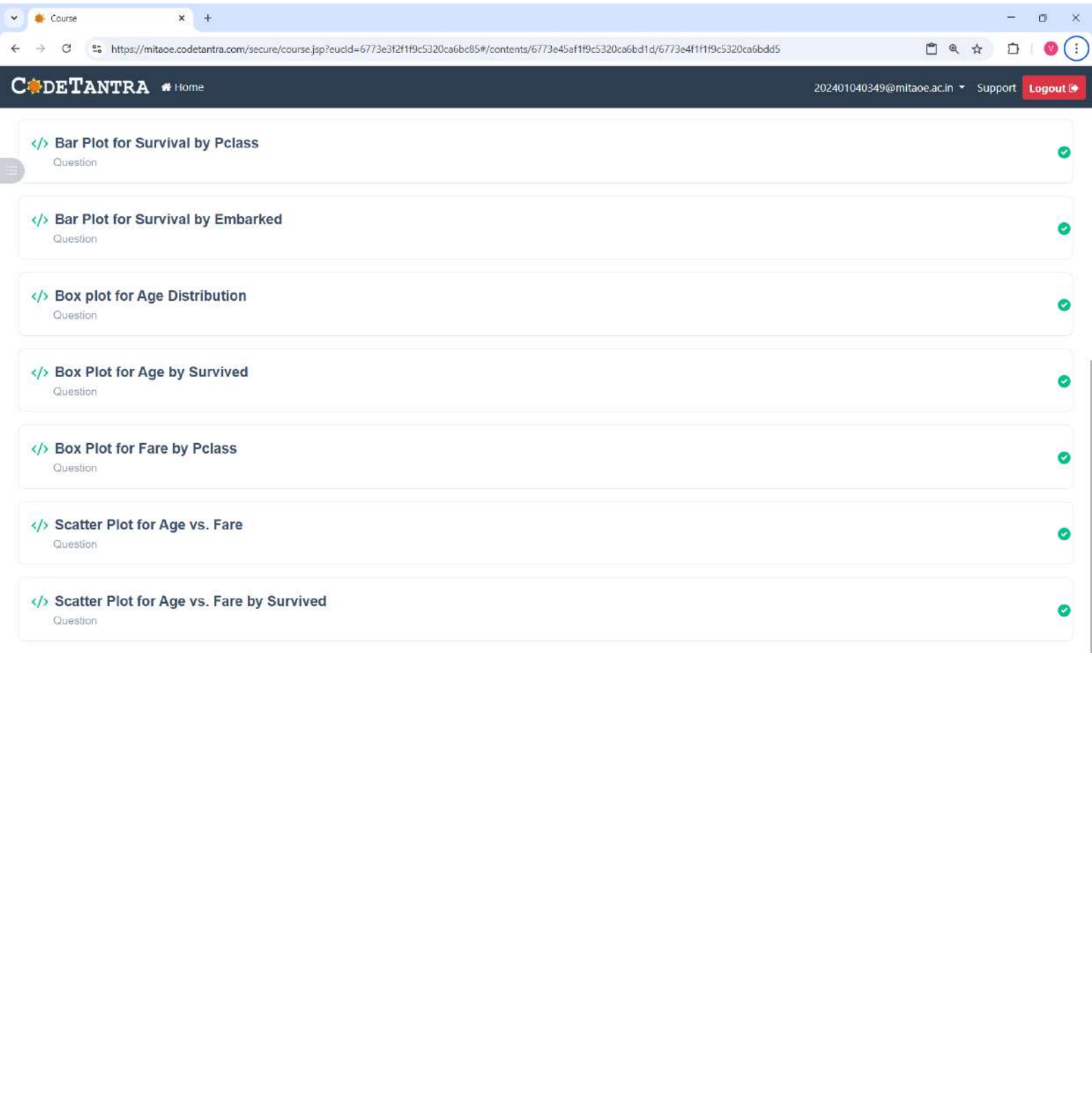
Debugger

```
    '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  }
11  #Create a dataframe from the data
12  df = pd.DataFrame(data)
13
14  #Creating stacked area plot
15  plt.stackplot(df['Month'],df['City_A_Temperature'],df['City_B_Tem
    perature'],df['City_C_Temperature'])
16
17  #Adding labels and titles
18  plt.xlabel('Month')
19  plt.ylabel('Temperature')
20  plt.title('Temperature Variation')
21
22  #Display plot
23  plt.show()
24
25
26
```

Terminal Test cases

< Prev Reset Submit Next >


Course
https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e45af1f9c5320ca6bd1d/6773e4f1f1f9c5320ca6bdd5
CODETANTRA
Home
202401040349@mitaoe.ac.in
Support
Logout
Lab Assignment
About this unit
Lab Assignment
Titanic Dataset
Question
Histogram of passenger information of Titanic
Question
Bar plot of survival rate of passengers
Question
Bar Plot for Survival by Gender
Question
Bar Plot for Survival by Pclass
Question


Course
https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e45af1f9c5320ca6bd1d/6773e4f1f1f9c5320ca6bdd5
CODETANTRA
Home
202401040349@mitaoe.ac.in
Support
Logout
Bar Plot for Survival by Pclass
Question
Bar Plot for Survival by Embarked
Question
Box plot for Age Distribution
Question
Box Plot for Age by Survived
Question
Box Plot for Fare by Pclass
Question
Scatter Plot for Age vs. Fare
Question
Scatter Plot for Age vs. Fare by Survived
Question

Course

https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e45af1f9c5320ca6bd1d/6773e4f1f1f9c5320ca6bdd5/677cbdd52372a2277b32ab1c

CODETANTRA Home 202401040349@mitaoe.ac.in Support Logout

## 5.2.1. Titanic Dataset

30:11

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 +

Explorer

titanicDat...

Submit

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 #Bar plot
11 pclass_counts = df['Pclass'].value_counts()
12 axes[0,0].bar(pclass_counts.index,pclass_counts.values, color =
   'skyblue')
13 axes[0,0].set_title('Passenger Class Distribution')
14 axes[0,0].set_xlabel('Pclass')
15 axes[0,0].set_ylabel('Count')
16
17 #Pie chart
18 gender_counts = df['Gender'].value_counts()
19 axes[0,1].pie(gender_counts,labels=gender_counts.index, autopct
   = '%1.1f%%', colors = ['lightblue','lightcoral'])
20 axes[0,1].set_title('Gender Distribution')
21
22 #Histogram
23 axes[1,0].hist(df['Age'],bins=8,color='lightgreen',edgecolor='bla
   ck')
```

Terminal Test cases

< Prev Reset Submit Next >

Course

https://mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e45af1f9c5320ca6bd1d/6773e4f1f1f9c5320ca6bdd5/677cbdd52372a2277b32ab1c

CODETANTRA Home 202401040349@mitaoe.ac.in Support Logout

## 5.2.1. Titanic Dataset

30:11

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 +

Explorer

titanicDat...

Submit

Debugger

```
22  #Histogram
23  axes[1,0].hist(df['Age'],bins=8,color='lightgreen',edgecolor='bla
    ck')
24  axes[1,0].set_title('Age Distribution')
25  axes[1,0].set_xlabel('Age')
26  axes[1,0].set_ylabel('Frequency')
27
28  #Countplot
29  survival_counts = df['Survived'].value_counts()
30  axes[1,1].bar(survival_counts.index,survival_counts.values,color=
    ['lightblue','lightcoral'])
31  axes[1,1].set_title('Survival Count')
32  axes[1,1].set_xlabel('Age')
33  axes[1,1].set_ylabel('Count')
34
35  #Scatter plot
36  axes[2,0].scatter(df['Age'],df['Fare'],color='orange')
37  axes[2,0].set_title('Fare vs Age')
38  axes[2,0].set_xlabel('Age')
39  axes[2,0].set_ylabel('Fare')
40
41  #Adjust layout to avoid overlap
42  plt.tight_layout()
43
44  #Show the Plot
45  plt.show()
```

Terminal Test cases

< Prev Reset Submit Next >

## 5.2.2. Histogram of passenger information of Titanic

03:25

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
```

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
12 # Convert categorical features to numeric
13 data['Sex'] = data['Sex'].map({'male': 0, 'female': 1})
14 data = pd.get_dummies(data, columns=['Embarked'],
   drop_first=True)
15
16 # Histogram for age distribution
17 plt.figure()
18 plt.hist(data['Age'], bins=30, edgecolor ='k')
19 plt.title('Age Distribution')
20 plt.xlabel('Age')
21 plt.ylabel('Frequency')
22 plt.show()
```

Terminal Test cases

Debugger

Course

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

CODETANTRA Home 202401040349@mitaoe.ac.in Support Logout

## 5.2.3. Bar plot of survival rate of passengers

04:04

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,

| Passenger Id | 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

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
12 # Convert categorical features to numeric
13 data['Sex'] = data['Sex'].map({'male': 0, 'female': 1})
14 data = pd.get_dummies(data, columns=['Embarked'],
   drop_first=True)
15
16 # Write your code here for Bar Plot for Survival Rate
17 plt.figure()
18 data['Survived'].value_counts().plot(kind='bar')
19 plt.title('Survival Count')
20 plt.xlabel('Survived')
21 plt.ylabel('Count')
22 plt.show()
23
24
```

Terminal Test cases

< Prev Reset Submit Next >

Course

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

CODETANTRA Home 202401040349@mitaoe.ac.in Support Logout

## 5.2.4. Bar Plot for Survival by Gender

07:37

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,

| Passengerld | 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
12 # Convert categorical features to numeric
13 data['Sex'] = data['Sex'].map({'male': 0, 'female': 1})
14 data = pd.get_dummies(data, columns=['Embarked'],
   drop_first=True)
15
16 # Write your code here for Bar Plot for Survival by Gender
17 plt.figure()
18 data.groupby('Sex')
   ['Survived'].value_counts().unstack().plot(kind='bar', stacked =
   True)
19 plt.title('Survival by Gender')
20 plt.xlabel('Gender')
21 plt.ylabel('Count')
22 plt.legend(['Not Survived','Survived'])
23 plt.show()
```

Terminal Test cases

< Prev Reset Submit Next >

Course

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

CODETANTRA Home 202401040349@mitaoe.ac.in Support Logout

## 5.2.5. Bar Plot for Survival by Pclass

04:25

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,

| Passenger Id | Survived | Pclass | Name | Sex | Age | SibSp | Parch | Ticket | Fare | Cabin | Embarked |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | |

**Sample Data:**

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
12 # Convert categorical features to numeric
13 data['Sex'] = data['Sex'].map({'male': 0, 'female': 1})
14 data = pd.get_dummies(data, columns=['Embarked'],
   drop_first=True)
15
16 # Write your code here for Bar Plot for Survival by Pclass
17 plt.figure()
18 data.groupby('Pclass')
   ['Survived'].value_counts().unstack().plot(kind='bar',stacked=Tru
   e)
19 plt.title('Survival by Pclass')
20 plt.xlabel('Pclass')
21 plt.ylabel('Count')
22 plt.legend(['Not Survived', 'Survived'])
23 plt.show()
```

Terminal Test cases

Debugger

< Prev Reset Submit Next >

Course

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

CODETANTRA Home 202401040349@mitaoe.ac.in Support Logout

## 5.2.6. Bar Plot for Survival by Embarked

08:23

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
12 # Convert categorical features to numeric
13 data['Sex'] = data['Sex'].map({'male': 0, 'female': 1})
14 data = pd.get_dummies(data, columns=['Embarked'],
   drop_first=True)
15
16 # Write your code here for Bar Plot for Survival by Embarked
17 plt.figure()
18 data.groupby('Embarked_Q')
   ['Survived'].value_counts().unstack().plot(kind='bar',stacked=Tru
   e)
19 plt.title('Survival by Embarked')
20 plt.xlabel('Embarked')
21 plt.ylabel('Count')
22 plt.legend(['Not Survived','Survived'])
23 plt.show()
```

Terminal Test cases

< Prev Reset Submit Next >

Course

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

CODETANTRA Home 202401040349@mitaoe.ac.in Support Logout

## 5.2.7. Box plot for Age Distribution

02:41

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,

| 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
```

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
12 # Convert categorical features to numeric
13 data['Sex'] = data['Sex'].map({'male': 0, 'female': 1})
14 data = pd.get_dummies(data, columns=['Embarked'],
   drop_first=True)
15
16 # Write your code here for Box Plot for Age by Pclass
17 plt.figure()
18 data.boxplot(column='Age', by='Pclass')
19 plt.title('Age by Pclass')
20 plt.suptitle('')
21 plt.xlabel('Pclass')
22 plt.ylabel('Age')
23 plt.show()
24
25
```

Terminal Test cases

< Prev Reset Submit Next >

Course

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

CODETANTRA Home

202401040349@mitaoe.ac.in Support Logout

## 5.2.8. Box Plot for Age by Survived

02:26

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,

| 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
```

Sample Test Cases +

Explorer

BoxPlotF...

Submit

Debugger

```python
import pandas as pd
import matplotlib.pyplot as plt

# Load the Titanic dataset
data = pd.read_csv('Titanic-Dataset.csv')

# Data Cleaning
data['Age'].fillna(data['Age'].median(), inplace=True)
data['Embarked'].fillna(data['Embarked'].mode()[0], inplace=True)
data.drop('Cabin', axis=1, inplace=True)

# Convert categorical features to numeric
data['Sex'] = data['Sex'].map({'male': 0, 'female': 1})
data = pd.get_dummies(data, columns=['Embarked'], drop_first=True)

# Write your code here for Box Plot for Age by Survived
plt.figure()
data.boxplot(column='Age', by='Survived')
plt.title('Age by Survival')
plt.suptitle('')
plt.xlabel('Survived')
plt.ylabel('Age')
plt.show()
```

Terminal Test cases

< Prev Reset Submit Next >

Course

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

CODETANTRA Home 202401040349@mitaoe.ac.in Support Logout

## 5.2.9. Box Plot for Fare by Pclass

02:30

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,

| Passenger Id | 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
12 # Convert categorical features to numeric
13 data['Sex'] = data['Sex'].map({'male': 0, 'female': 1})
14 data = pd.get_dummies(data, columns=['Embarked'],
   drop_first=True)
15
16 # Write your code here for Box Plot for Fare by Pclass
17 plt.figure()
18 data.boxplot(column='Fare', by = 'Pclass')
19 plt.title('Fare by Pclass')
20 plt.suptitle('')
21 plt.xlabel('Pclass')
22 plt.ylabel('Fare')
23 plt.show()
24
25
```

Debugger

Terminal Test cases

< Prev Reset Submit Next >

Course

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

CODETANTRA Home 202401040349@mitaoe.ac.in Support Logout

## 5.2.10. Scatter Plot for Age vs. Fare

02:51

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,

| 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 17599,71.283
```

Sample Test Cases +

Explorer

AgeFareS...

Submit

```python
import pandas as pd
import matplotlib.pyplot as plt

# Load the Titanic dataset
data = pd.read_csv('Titanic-Dataset.csv')

# Data Cleaning
data['Age'].fillna(data['Age'].median(), inplace=True)
data['Embarked'].fillna(data['Embarked'].mode()[0], inplace=True)
data.drop('Cabin', axis=1, inplace=True)

# Convert categorical features to numeric
data['Sex'] = data['Sex'].map({'male': 0, 'female': 1})
data = pd.get_dummies(data, columns=['Embarked'], drop_first=True)

# Write your code here for Box Plot for Fare by Pclass
plt.figure()
plt.scatter(data['Age'], data['Fare'])
plt.title('Age vs. Fare')
plt.xlabel('Age')
plt.ylabel('Fare')
plt.show()
```

Debugger

Terminal Test cases

< Prev Reset Submit Next >

Course

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

CODETANTRA Home 202401040349@mitaoe.ac.in Support Logout

## 5.2.11. Scatter Plot for Age vs. Fare by Survived

03:59

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,

| 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
```

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
12 # Convert categorical features to numeric
13 data['Sex'] = data['Sex'].map({'male': 0, 'female': 1})
14 data = pd.get_dummies(data, columns=['Embarked'],
   drop_first=True)
15
16 # Write your code here for Scatter Plot for Age vs. Fare by
   Survived
17 plt.figure()
18 colors = {0:'red',1:'blue'}
19 plt.scatter(data['Age'],data['Fare'],c=data['Survived'].apply(lam
   bda x: colors[x]))
20 plt.title('Age vs. Fare by Survival')
21 plt.xlabel('Age')
22 plt.ylabel('Fare')
23 plt.show()
```

Terminal Test cases

‹ Prev Reset Submit