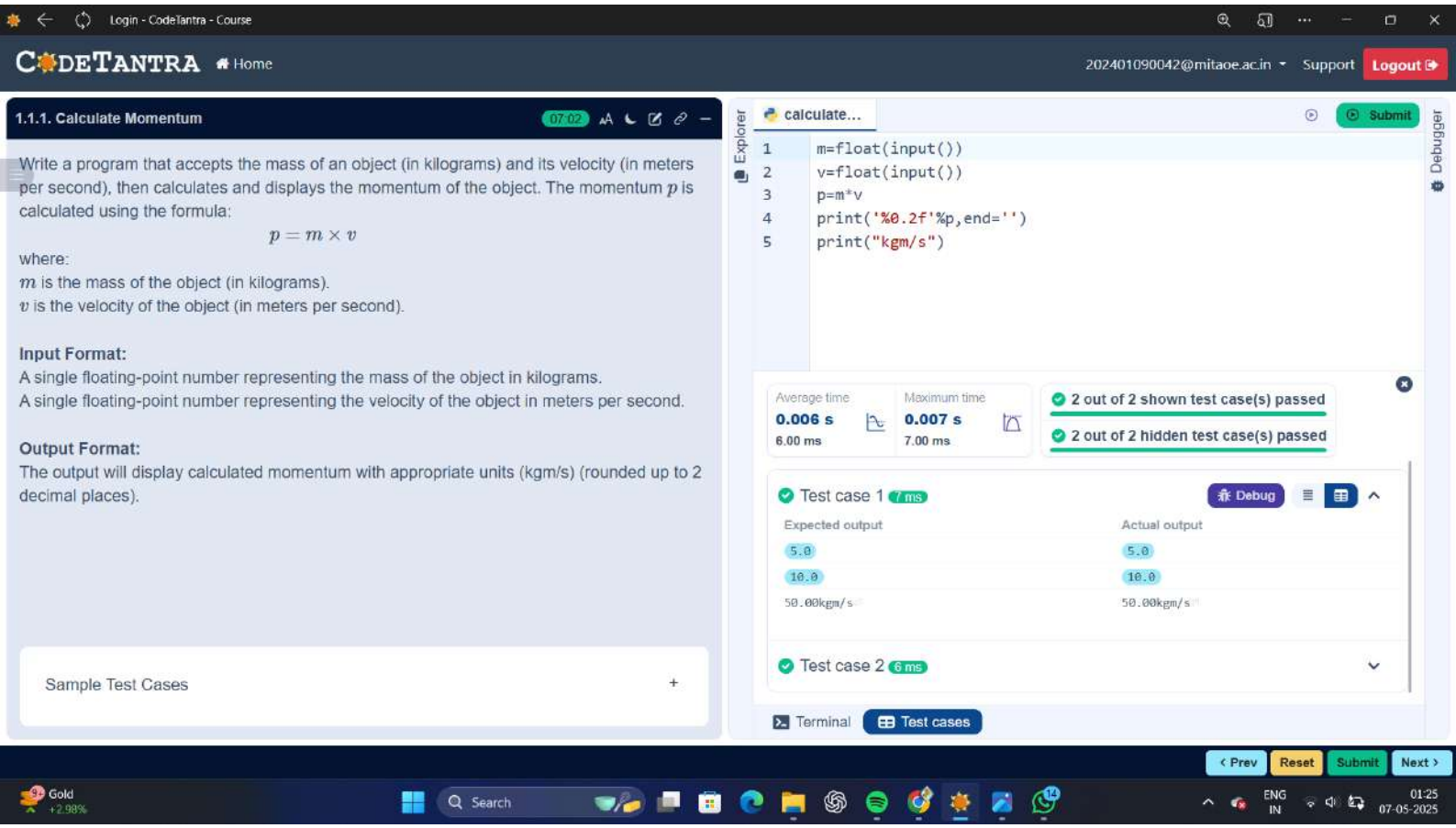
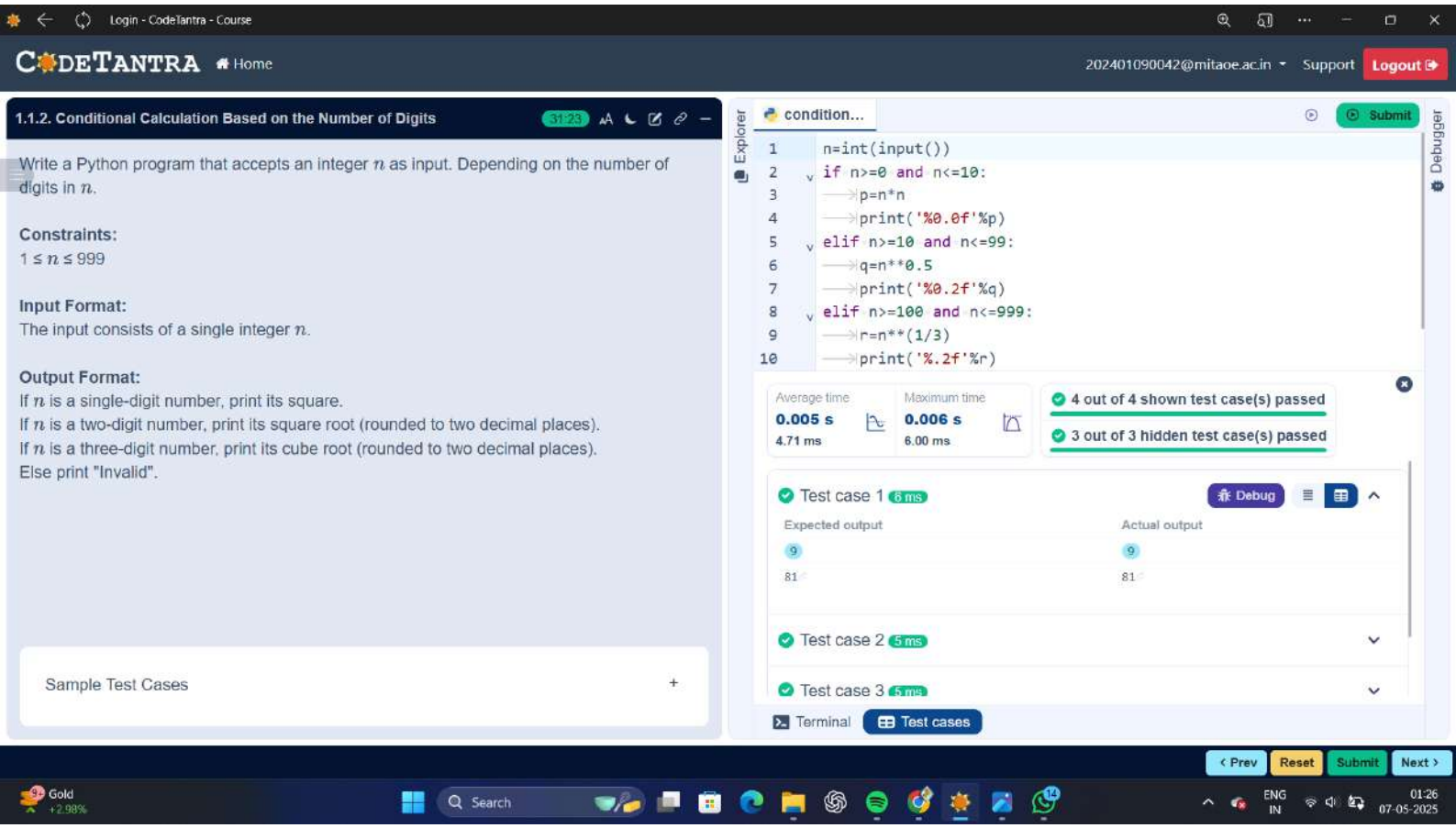
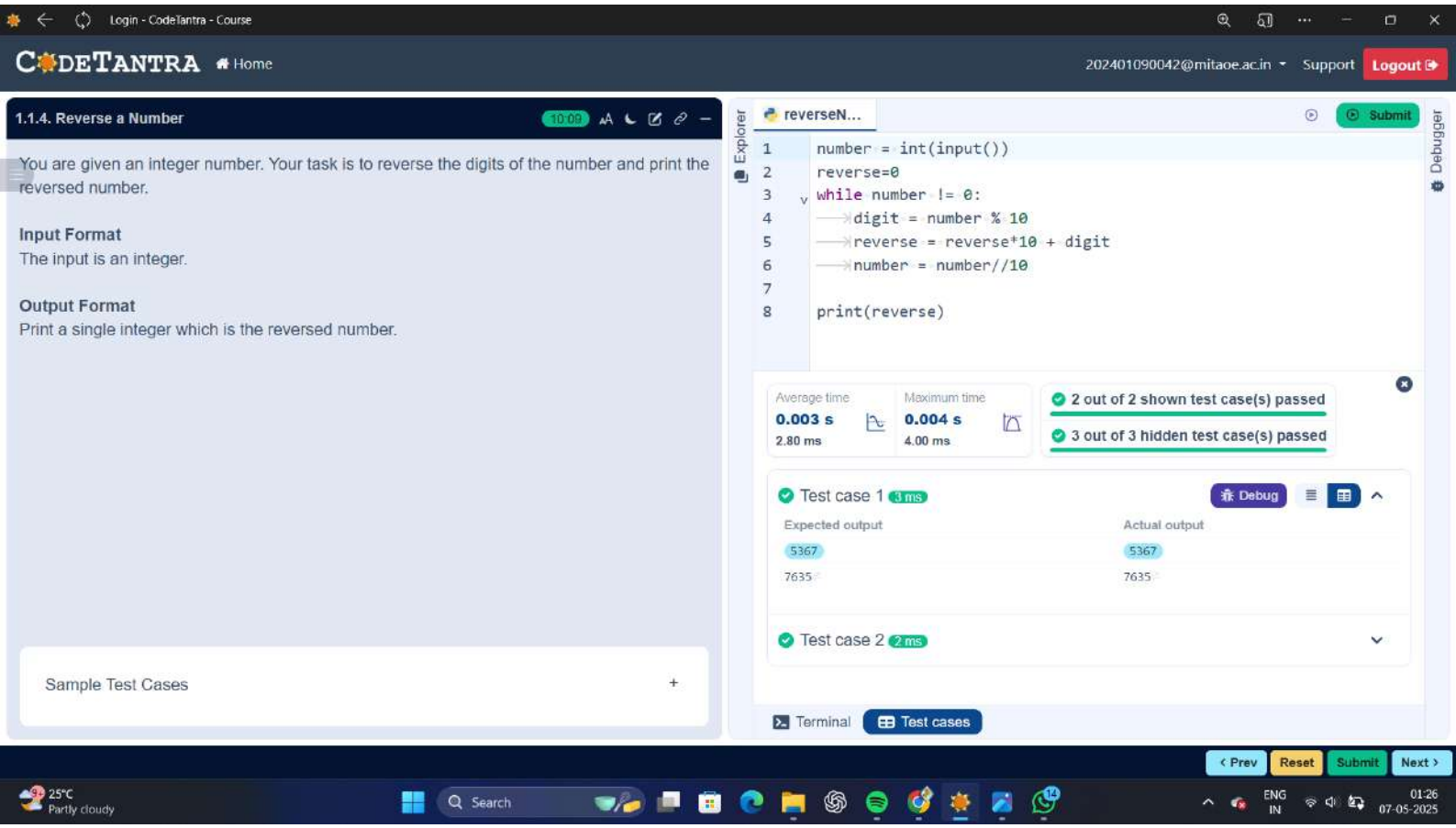
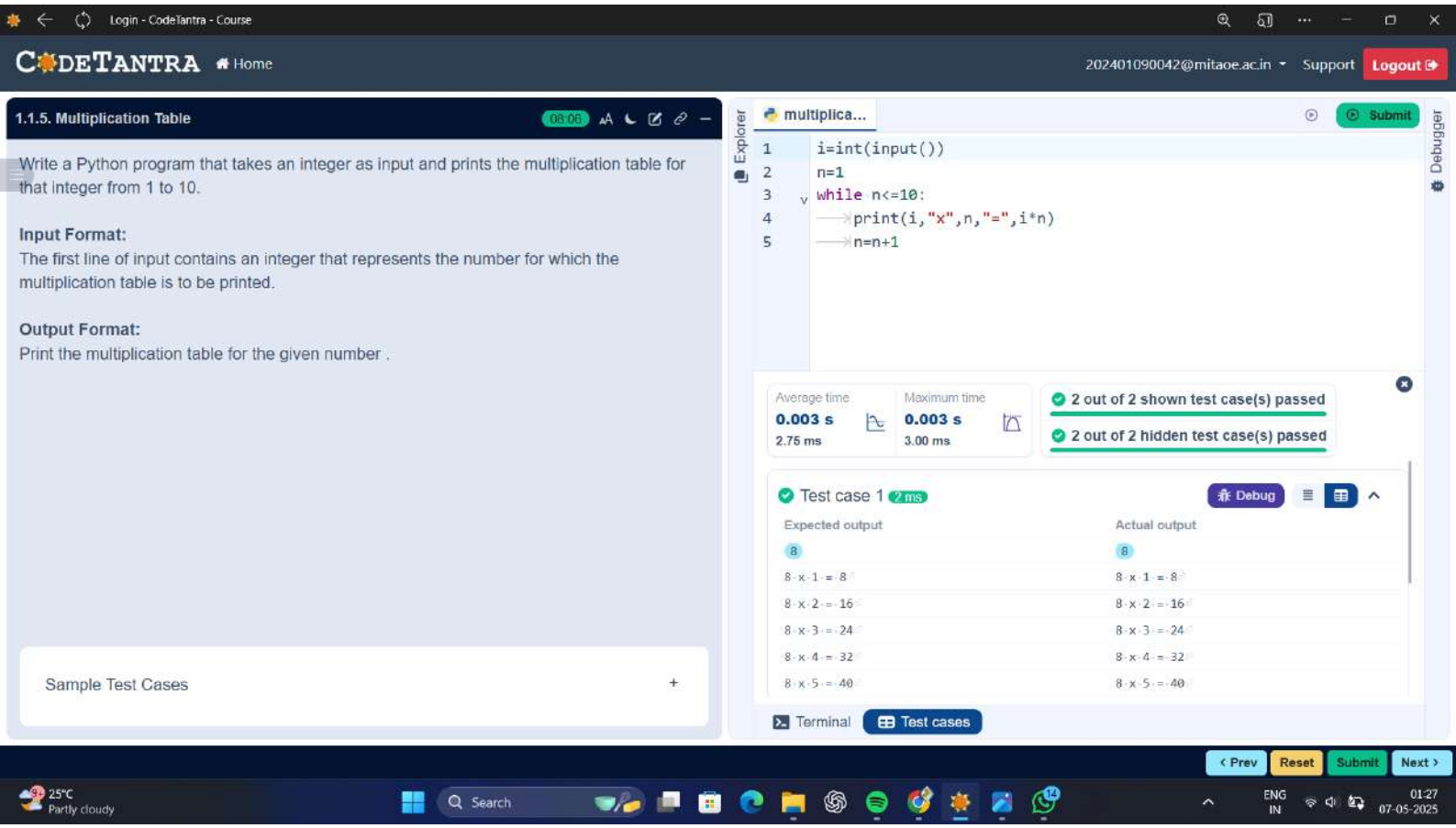
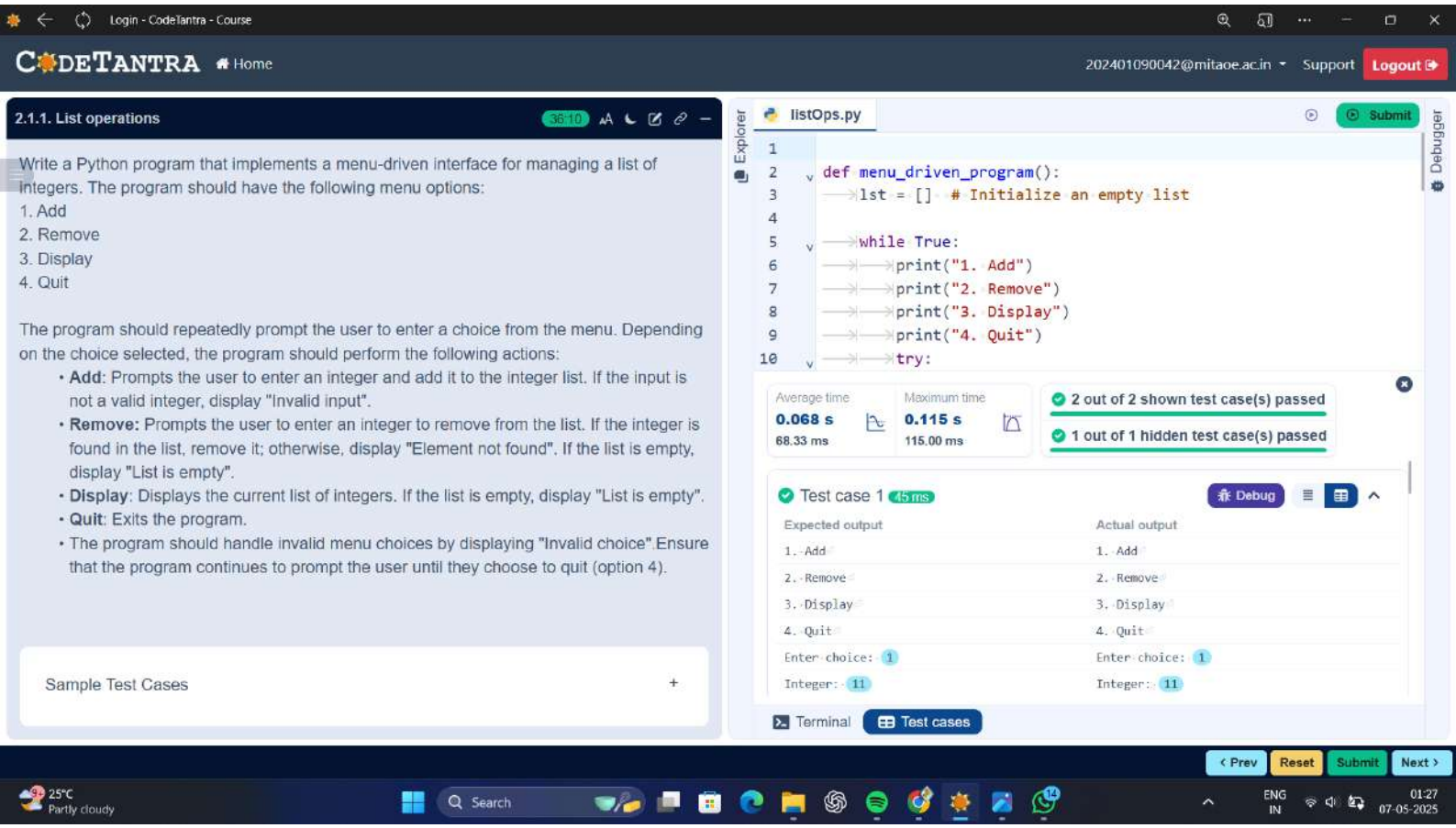
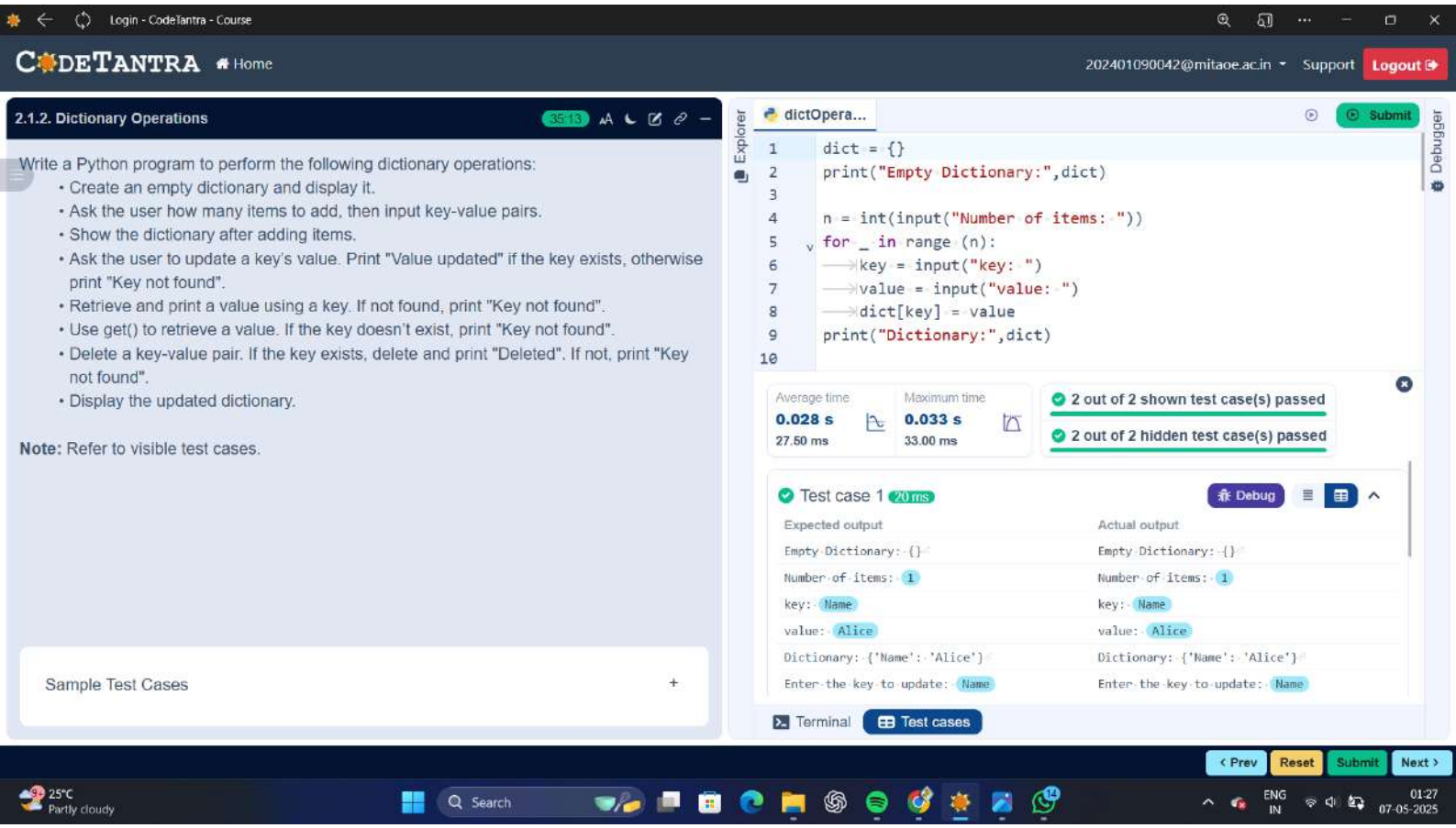

Login - Codetantra - Course
CODETANTRA
Home
202401090042@mitaoe.ac.in
Support
Logout
1.1.1. Calculate Momentum
07:02
Write a program that accepts the mass of an object (in kilograms) and its velocity (in meters per second), then calculates and displays the momentum of the object. The momentum p is calculated 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('%0.2f'%p,end='')
5 print("kgm/s")
Average time
0.006 s
6.00 ms
Maximum time
0.007 s
7.00 ms
2 out of 2 shown test case(s) passed
2 out of 2 hidden test case(s) passed
Test case 1 7 ms
Debug
Expected output
5.0
10.0
50.00kgm/s
Actual output
5.0
10.0
50.00kgm/s
Test case 2 6 ms
Terminal
Test cases
Prev
Reset
Submit
Next
Gold +2.98%
Search
ENG IN
01:25
07-05-2025


Login - Codetantra - Course
CODETANTRA
Home
202401090042@mitaoe.ac.in
Support
Logout
1.1.2. Conditional Calculation Based on the Number of Digits
31:23
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 n>=0 and n<=10:
3 p=n*n
4 print('%0.0f'%p)
5 elif n>=10 and n<=99:
6 q=n**0.5
7 print('%0.2f'%q)
8 elif n>=100 and n<=999:
9 r=n**(1/3)
10 print('%.2f'%r)
Average time
0.005 s
4.71 ms
Maximum time
0.006 s
6.00 ms
4 out of 4 shown test case(s) passed
3 out of 3 hidden test case(s) passed
Test case 1 6 ms
Debug
Expected output
9
81
Actual output
9
81
Test case 2 5 ms
Test case 3 5 ms
Terminal
Test cases
Prev
Reset
Submit
Next
Gold
+2.98%
Search
ENG
IN
01:26
07-05-2025


Login - Codetantra - Course
CODETANTRA
Home
202401090042@mitaoe.ac.in
Support
Logout
1.1.4. Reverse a Number
10:00
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 number = int(input())
2 reverse=0
3 while number != 0:
4 digit = number % 10
5 reverse = reverse*10 + digit
6 number = number//10
7
8 print(reverse)
Average time
0.003 s
2.80 ms
Maximum time
0.004 s
4.00 ms
2 out of 2 shown test case(s) passed
3 out of 3 hidden test case(s) passed
Test case 1 4 ms
Debug
Expected output
5367
7635
Actual output
5367
7635
Test case 2 2 ms
Terminal
Test cases
Prev
Reset
Submit
Next
25°C
Partly cloudy
Search
ENG
IN
01:26
07-05-2025


Login - Codetantra - Course
CODETANTRA
Home
202401090042@mitaoe.ac.in
Support
Logout
1.1.5. Multiplication Table
08:06
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 i=int(input())
2 n=1
3 while n<=10:
4 print(i,"x",n,"=",i*n)
5 n=n+1
Average time
0.003 s
2.75 ms
Maximum time
0.003 s
3.00 ms
2 out of 2 shown test case(s) passed
2 out of 2 hidden test case(s) passed
Test case 1
7 ms
Debug
Expected output
Actual output
8
8 x 1 = 8
8 x 2 = 16
8 x 3 = 24
8 x 4 = 32
8 x 5 = 40
Terminal
Test cases
Prev
Reset
Submit
Next
25°C
Partly cloudy
Search
ENG
IN
01:27
07-05-2025


Login - Codelantra - Course
CODETANTRA
Home
202401090042@mitaoe.ac.in
Support
Logout
2.1.1. List operations
36:10
Write a Python program that implements a menu-driven interface for managing a list of integers. The program should have the following menu options:
1. Add
2. Remove
3. Display
4. Quit
The program should repeatedly prompt the user to enter a choice from the menu. Depending on the choice selected, the program should perform the following actions:
• Add: Prompts the user to enter an integer and add it to the integer list. If the input is not a valid integer, display "Invalid input".
• Remove: Prompts the user to enter an integer to remove from the list. If the integer is found in the list, remove it; otherwise, display "Element not found". If the list is empty, display "List is empty".
• Display: Displays the current list of integers. If the list is empty, display "List is empty".
• Quit: Exits the program.
• The program should handle invalid menu choices by displaying "Invalid choice".Ensure that the program continues to prompt the user until they choose to quit (option 4).
Sample Test Cases
Explorer
listOps.py
Submit
Debugger
def menu_driven_program():
lst = [] # Initialize an empty list
while True:
print("1. Add")
print("2. Remove")
print("3. Display")
print("4. Quit")
try:
Average time
0.068 s
68.33 ms
Maximum time
0.115 s
115.00 ms
2 out of 2 shown test case(s) passed
1 out of 1 hidden test case(s) passed
Test case 1 45 ms
Debug
Expected output
Actual output
1. Add
2. Remove
3. Display
4. Quit
Enter choice: 1
Integer: 11
Terminal
Test cases
Prev
Reset
Submit
Next
25°C
Partly cloudy
Search
ENG
IN
01:27
07-05-2025


Login - Codelantra - Course
CODETANTRA
Home
202401090042@mitaoe.ac.in
Support
Logout
2.1.2. Dictionary Operations
35:13
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 dict = {}
2 print("Empty Dictionary:",dict)
3
4 n = int(input("Number of items: "))
5 for _ in range (n):
6 key = input("key: ")
7 value = input("value: ")
8 dict[key] = value
9 print("Dictionary:",dict)
10
Average time
0.028 s
27.50 ms
Maximum time
0.033 s
33.00 ms
2 out of 2 shown test case(s) passed
2 out of 2 hidden test case(s) passed
Test case 1 20 ms
Debug
Expected output
Empty Dictionary: {}
Number of items: 1
key: Name
value: Alice
Dictionary: {'Name': 'Alice'}
Enter the key to update: Name
Actual output
Empty Dictionary: {}
Number of items: 1
key: Name
value: Alice
Dictionary: {'Name': 'Alice'}
Enter the key to update: Name
Terminal
Test cases
‹ Prev
Reset
Submit
Next ›
25°C
Partly cloudy
Search
ENG
IN
01:27
07-05-2025

Login - Codetantra - Course

CODETANTRA Home

202401090042@mitaoe.ac.in Support Logout

3.1.1. Numpy array operations

31:00

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  rows,cols= list(map(int,input().split()))
3  matrix= []
4  for i in range(rows):
5      row = list(map(int,input().split()))
6      matrix.append(row)
7  matrix= np.array(matrix).reshape(rows,cols)
8
9  print(matrix)
10 print(matrix.ndim)
```

Average time 0.009 s 8.60 ms

Maximum time 0.014 s 14.00 ms

3 out of 3 shown test case(s) passed

2 out of 2 hidden test case(s) passed

Test case 1 14 ms Debug

Expected output	Actual output
3 4	3 4
1 2 3 4	1 2 3 4
5 6 7 8	5 6 7 8
9 10 11 12	9 10 11 12
[[1 2 3 4]	[[1 2 3 4]
[5 6 7 8]	[5 6 7 8]

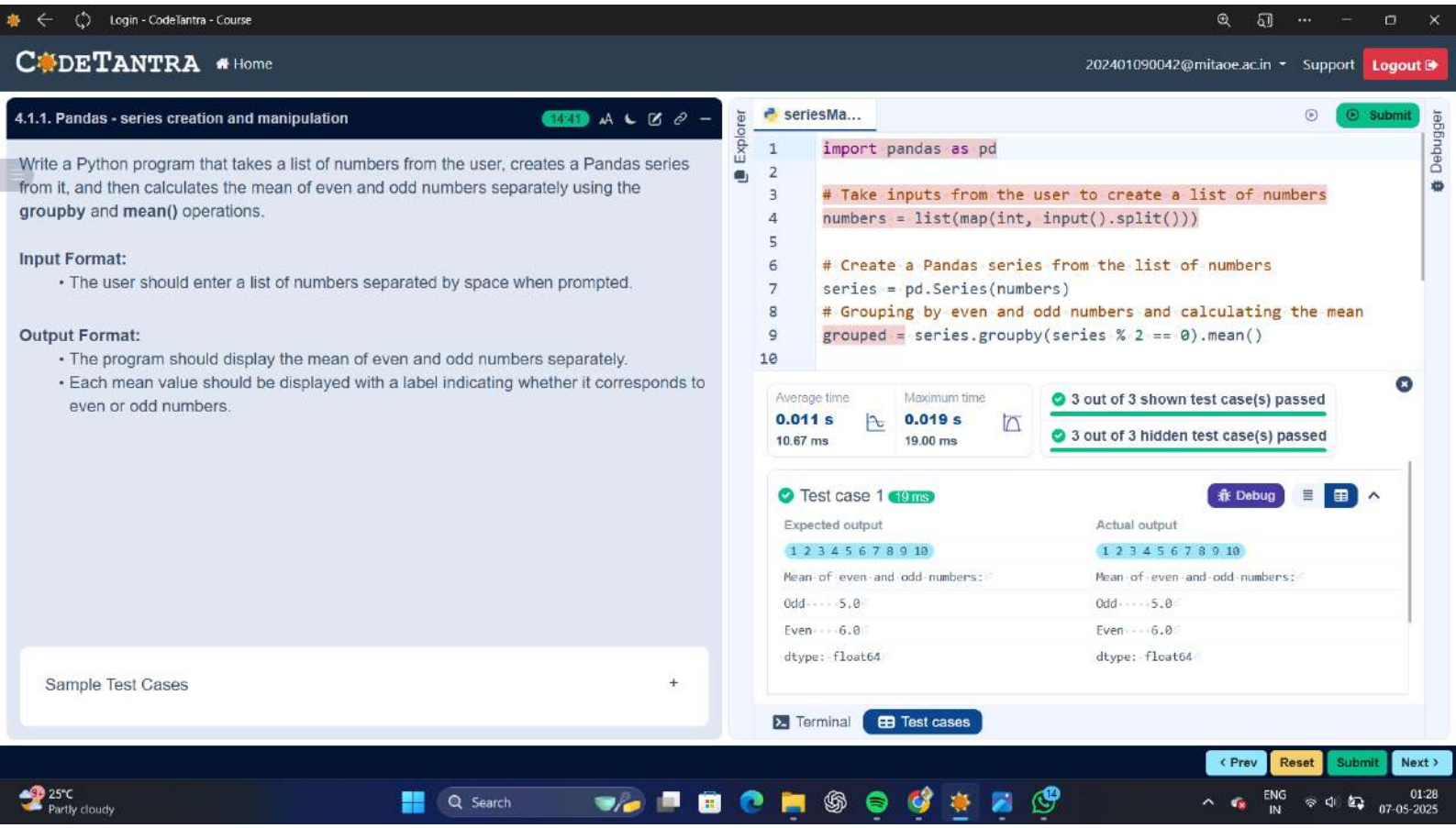
Terminal Test cases

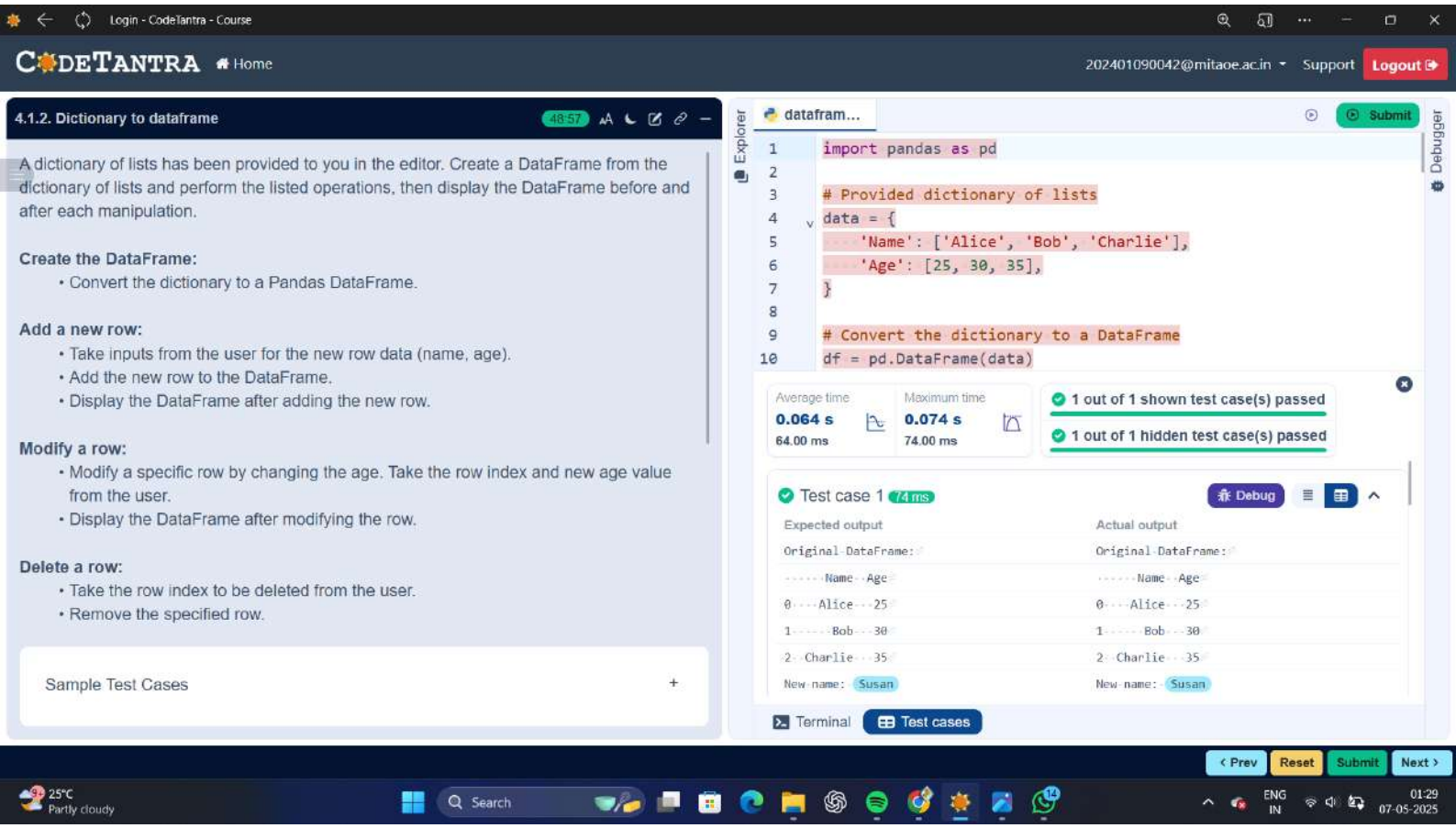
‹ Prev Reset Submit Next ›

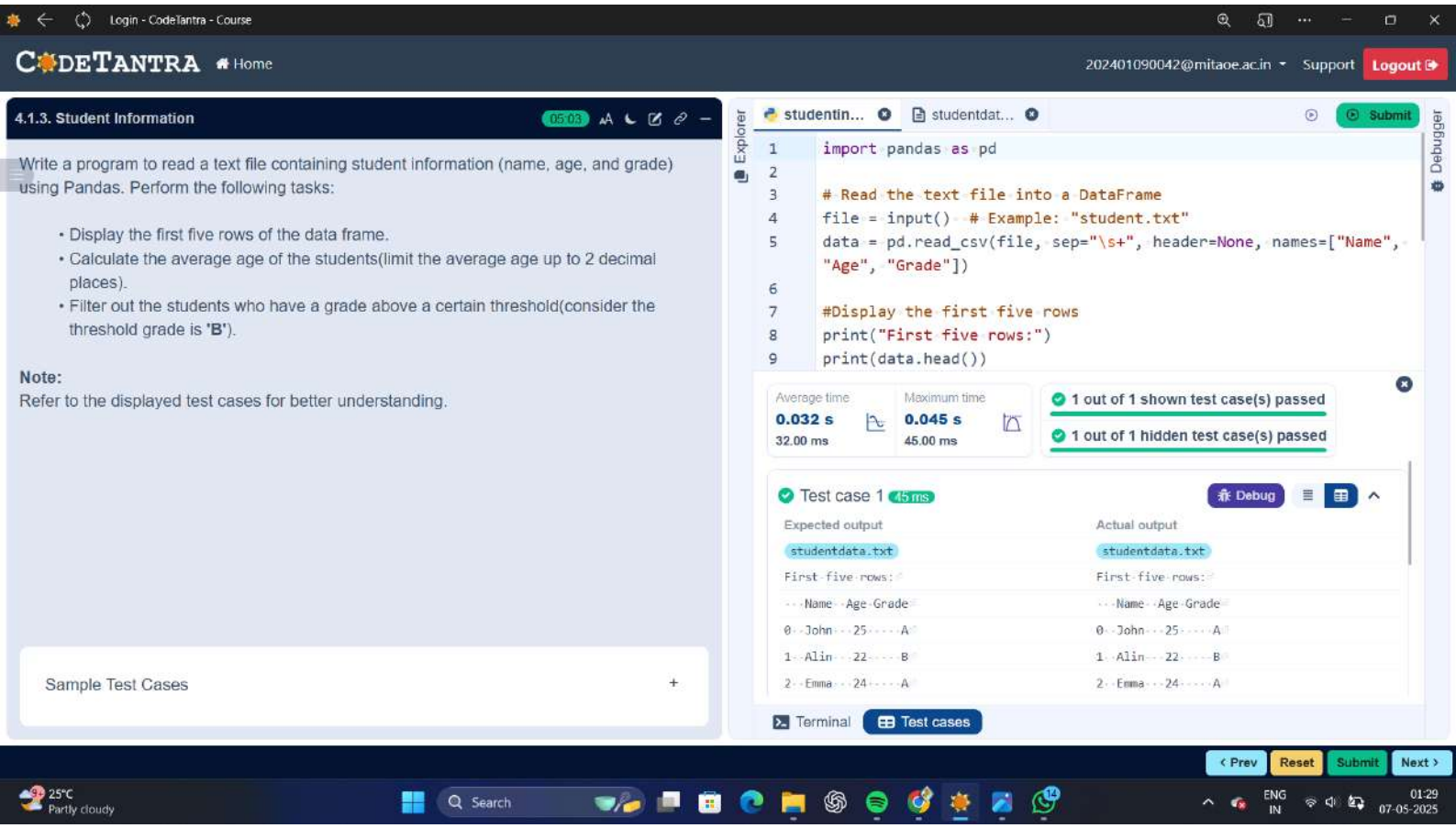
25°C Partly cloudy

Search

ENG IN 01:28 07-05-2025


Login - Codetantra - Course
CODETANTRA
Home
202401090042@mitaoe.ac.in
Support
Logout
4.1.1. Pandas - series creation and manipulation
14:41
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 series = pd.Series(numbers)
8 # Grouping by even and odd numbers and calculating the mean
9 grouped = series.groupby(series % 2 == 0).mean()
10
Average time
0.011 s
10.67 ms
Maximum time
0.019 s
19.00 ms
3 out of 3 shown test case(s) passed
3 out of 3 hidden test case(s) passed
Test case 1
19 ms
Debug
Expected output
1 2 3 4 5 6 7 8 9 10
Mean of even and odd numbers:
Odd 5.0
Even 6.0
dtype: float64
Actual output
1 2 3 4 5 6 7 8 9 10
Mean of even and odd numbers:
Odd 5.0
Even 6.0
dtype: float64
Terminal
Test cases
Prev
Reset
Submit
Next
25°C
Partly cloudy
Search
ENG
IN
01:28
07-05-2025


Login - Codelantra - Course
CODETANTRA
Home
202401090042@mitaoe.ac.in
Support
Logout
4.1.2. Dictionary to dataframe
48:57
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.
Sample Test Cases
Explorer
dataframe...
Submit
Debugger
import pandas as pd
# Provided dictionary of lists
data = {
'Name': ['Alice', 'Bob', 'Charlie'],
'Age': [25, 30, 35],
}
# Convert the dictionary to a DataFrame
df = pd.DataFrame(data)
Average time
0.064 s
64.00 ms
Maximum time
0.074 s
74.00 ms
1 out of 1 shown test case(s) passed
1 out of 1 hidden test case(s) passed
Test case 1
74 ms
Debug
Expected output
Original DataFrame:
Name Age
0 Alice 25
1 Bob 30
2 Charlie 35
New name: Susan
Actual output
Original DataFrame:
Name Age
0 Alice 25
1 Bob 30
2 Charlie 35
New name: Susan
Terminal
Test cases
Prev
Reset
Submit
Next
25°C
Partly cloudy
Search
ENG
IN
01:29
07-05-2025


Login - Codelantra - Course
CODETANTRA
Home
202401090042@mitaoe.ac.in
Support
Logout
4.1.3. Student Information
05:03
Write a program to read a text file containing student information (name, age, and grade) using Pandas. Perform the following tasks:
• Display the first five rows of the data frame.
• Calculate the average age of the students(limit the average age up to 2 decimal places).
• Filter out the students who have a grade above a certain threshold(consider the threshold grade is 'B').
Note:
Refer to the displayed test cases for better understanding.
Sample Test Cases
studentin...
studentdat...
Submit
Explorer
Debugger
1 import pandas as pd
2
3 # Read the text file into a DataFrame
4 file = input() # Example: "student.txt"
5 data = pd.read_csv(file, sep="\s+", header=None, names=["Name", "Age", "Grade"])
6
7 #Display the first five rows
8 print("First five rows:")
9 print(data.head())
Average time
0.032 s
32.00 ms
Maximum time
0.045 s
45.00 ms
1 out of 1 shown test case(s) passed
1 out of 1 hidden test case(s) passed
Test case 1
45 ms
Debug
Expected output
studentdata.txt
First five rows:
Name Age Grade
0 John 25 A
1 Alin 22 B
2 Emma 24 A
Actual output
studentdata.txt
First five rows:
Name Age Grade
0 John 25 A
1 Alin 22 B
2 Emma 24 A
Terminal
Test cases
‹ Prev
Reset
Submit
Next ›
25°C
Partly cloudy
Search
ENG
IN
01:29
07-05-2025

Login - Codelantra - Course

CODETANTRA Home

202401090042@mitaoe.ac.in Support Logout

5.1.1. Stacked Plot

14:38

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
12 # Write your code...
13 plt.stackplot(data['Month'],data['City_A_Temperature'],data['City
   _B_Temperature'],data['City_C_Temperature'])
14 plt.xlabel('Month')
15 plt.ylabel('Temperature')
16 plt.title('Temperature Variation')
17 plt.show()
```

Terminal Test cases

‹ Prev Reset Submit Next ›

25°C Partly cloudy

Search

ENG IN 01:29 07-05-2025

Login - Codetantra - Course

CODETANTRA Home

202401090042@mitaoe.ac.in Support Logout

5.1.1. Stacked Plot

14:38

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],
```

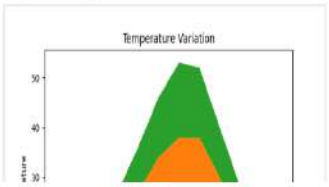
Average time 0.187 s 187.00 ms

Maximum time 0.187 s 187.00 ms

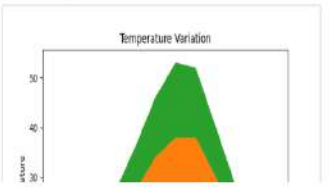
1 out of 1 shown test case(s) passed

Test case 1 187 ms Debug

Expected output



Actual output



Terminal Test cases

‹ Prev Reset Submit Next ›

25°C Partly cloudy

Search

ENG IN 01:30 07-05-2025