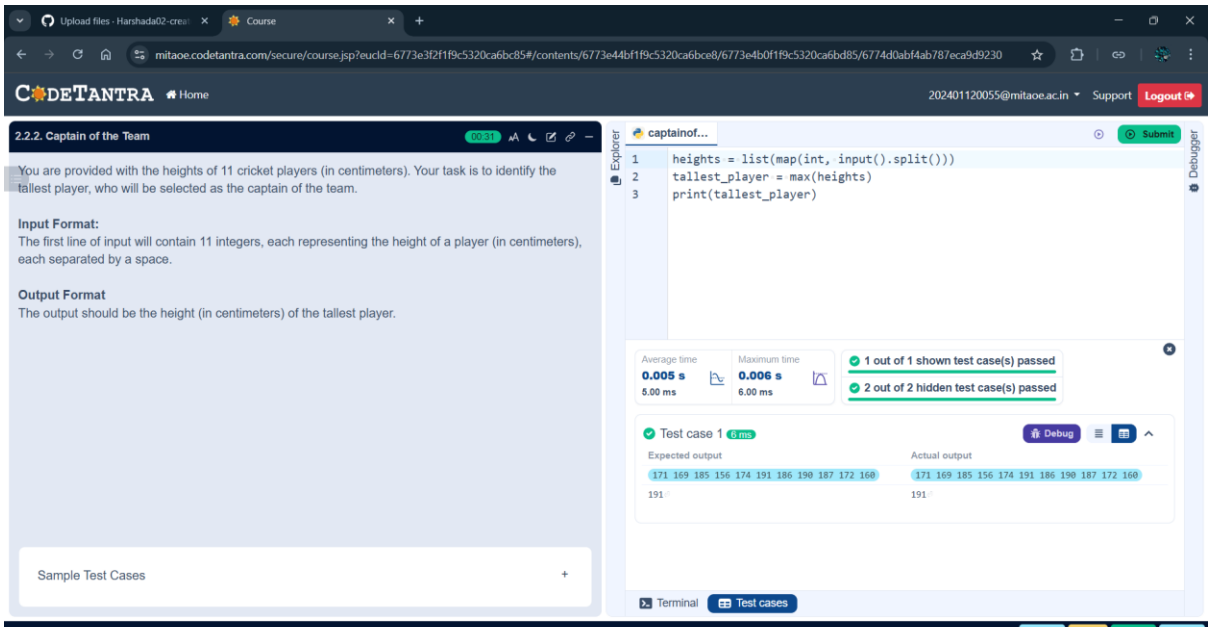
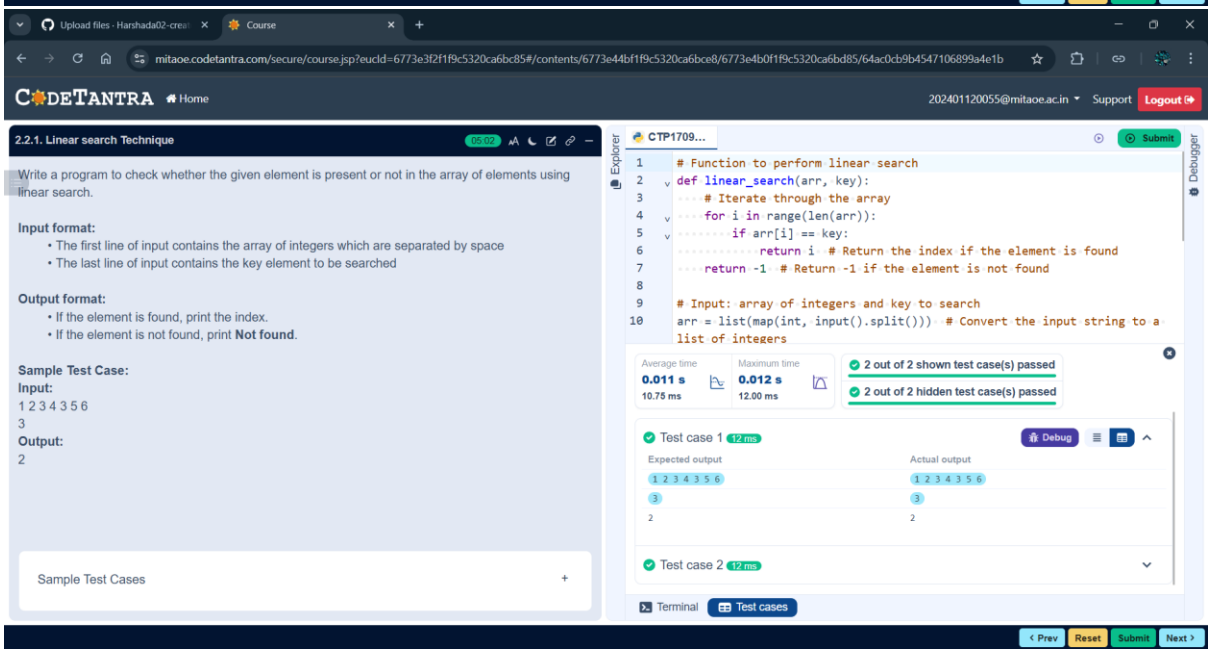
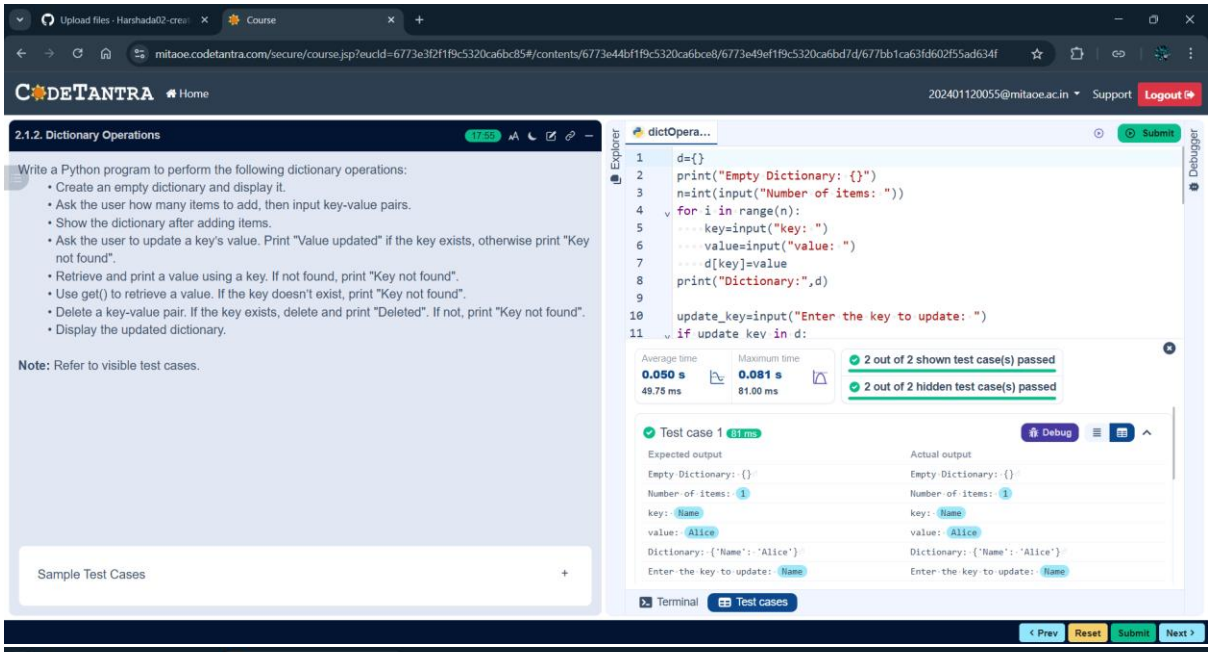
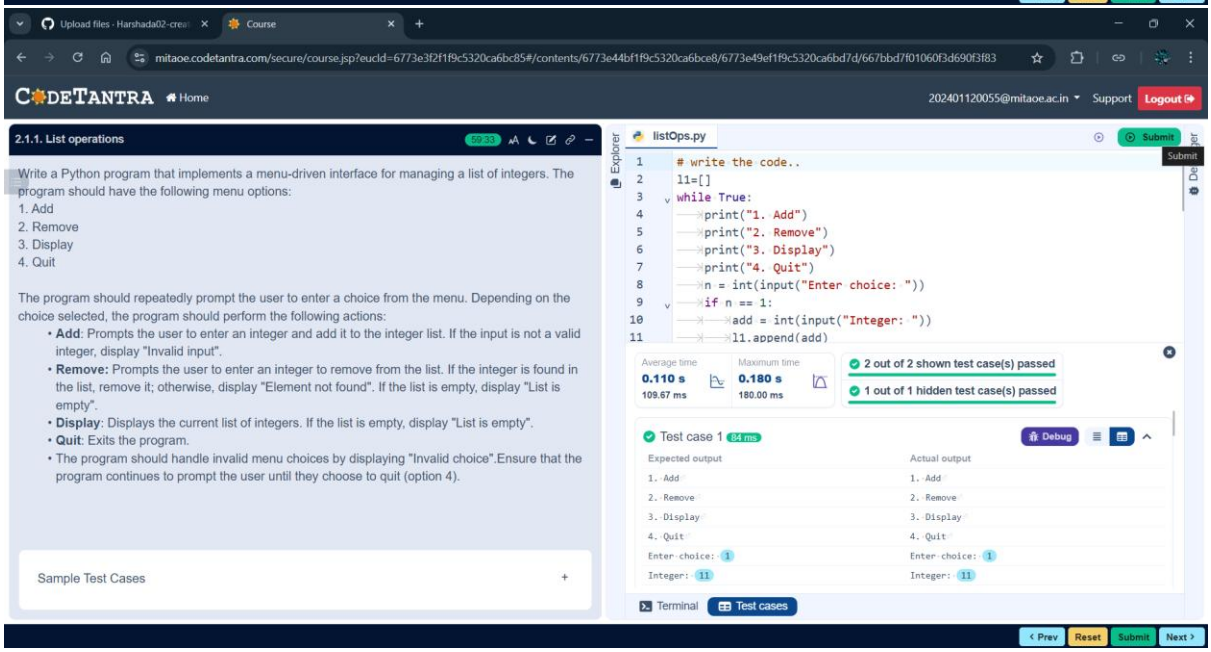
PRACTICAL 2

ALL PRACTICALS:






Upload files · Harshada02-crea
Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e44bf1f9c5320ca6bce8/6773e49ef1f9c5320ca6bd7d/677bb1ca63fd602f55ad634f
CODETANTRA
Home
202401120055@mitaoe.ac.in
Support
Logout
2.1.2. Dictionary Operations
17:55
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
d={}
print("Empty Dictionary: {}")
n=int(input("Number of items: "))
for i in range(n):
key=input("key: ")
value=input("value: ")
d[key]=value
print("Dictionary:",d)
update_key=input("Enter the key to update: ")
if update_key in d:
Average time
0.050 s
49.75 ms
Maximum time
0.081 s
81.00 ms
2 out of 2 shown test case(s) passed
2 out of 2 hidden test case(s) passed
Test case 1
81 ms
Debug
Expected output
Actual output
Empty Dictionary: {}
Number of items: 1
key: Name
value: Alice
Dictionary: {'Name': 'Alice'}
Enter the key to update: Name
Terminal
Test cases
‹ Prev
Reset
Submit
Next ›


Upload files · Harshada02-crea
Course
mitaoe.codetantra.com/secure/course.jsp?eucId=6773e3f2f1f9c5320ca6bc85#/contents/6773e44bf1f9c5320ca6bce8/6773e49ef1f9c5320ca6bd7d/667bbd7f01060f3d690f3f83
CODETANTRA
Home
202401120055@mitaoe.ac.in
Support
Logout
2.1.1. List operations
59:33
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
Submit
# write the code..
l1=[]
while True:
print("1. Add")
print("2. Remove")
print("3. Display")
print("4. Quit")
n = int(input("Enter choice: "))
if n == 1:
add = int(input("Integer: "))
l1.append(add)
Average time
0.110 s
109.67 ms
Maximum time
0.180 s
180.00 ms
2 out of 2 shown test case(s) passed
1 out of 1 hidden test case(s) passed
Test case 1
84 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 ›