2.1.1. List operations

Write a Python program that implements a menu-driven interface for managing a list of integers. The program should have the following menu options:

5610 A C Z 2

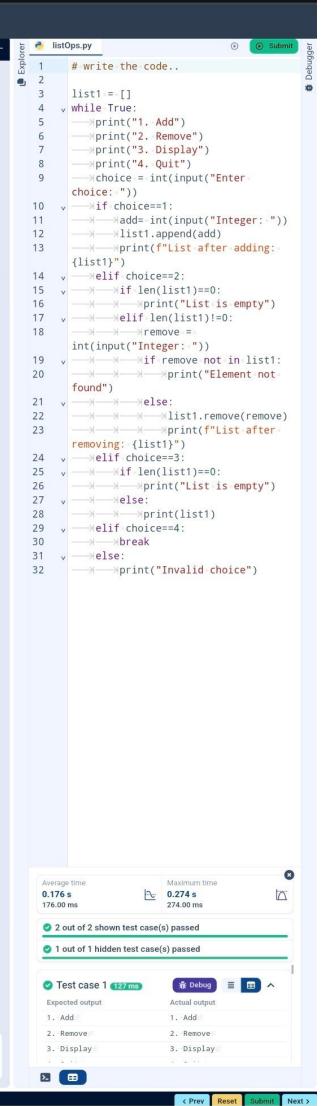
- 1. Add
- 2. Remove
- 3. Display
- 4. Quit

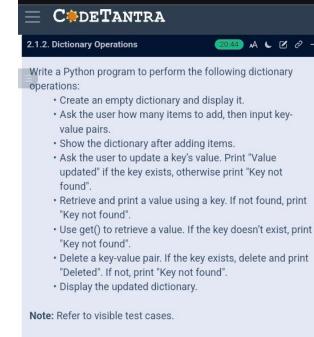
The program should repeatedly prompt the user to enter a choice from the menu. Depending on the choice selected, the program should perform the following actions:

- 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.

Sample Test Cases

 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

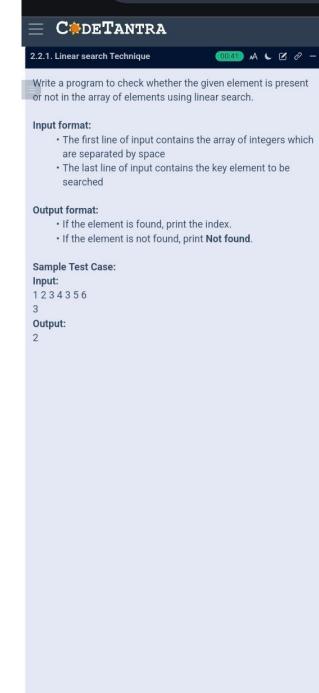
A L & 2 -

```
dictOperati...
                                         ( Submit
   1
         # Create an empty dictionary
   2
•
                                                 .
   3
         my_dict = {}
   4
         print(f"Empty Dictionary: {my_dict}")
   5
         # Get the number of items to add
   6
   7
         num_items = int(input("Number of
         items: "))
   8
   9
         # Add key-value pairs
  10
       v for _ in range(num_items):
  11
         key = input("key: ")
  12
          value = input("value: ")
  13
         my_dict[key] = value
  14
  15
         print(f"Dictionary: {my_dict}")
  17
         # Update a key's value
  18
         update_key = input("Enter the key to
         update: ")
  19
       v if update_key in my_dict:
  20
           new_value = input("Enter the new
         value: ")
  21
         my_dict[update_key] = new_value
  22
         print("Value updated")
  23
  24
         print("Key not found")
  25
  26
         # Retrieve a value using a key
  27
         retrieve_key = input("Enter the key
         to retrieve: ")
  28
       v if retrieve_key in my_dict:
         print(f"Key: {retrieve_key},
  29
         Value: {my_dict[retrieve_key]}")
  30
       v else:
  31
         print("Key not found")
  32
  33
         # Retrieve value using get()
  34
         get_key = input("Enter the key to
         get using the get() method: ")
  35
         value = my_dict.get(get_key, "Key
         not found")
       v if value != "Key not found":
  36
  37
          print(f"Key: {get_key}, Value:
         {value}")
  38
       v else:
  39
         print(value)
  40
  41
         # Delete a key-value pair
  42
         delete_key = input("Enter the key to
        delete: ")
       v if delete_key in my_dict:
  43
        del my dict[delete key]
  45
         print("Deleted")
  46
      v else:
  47
        print("Key not found")
  48
  49
       # Display the updated dictionary
    Average time
                         Maximum time
                     0.264 s
    0.153 s
                                            153.50 ms
                         264.00 ms
    2 out of 2 shown test case(s) passed
    2 out of 2 hidden test case(s) passed
    Test case 1 79 ms
                          🛣 Debug
     Expected output
                          Actual output
     Empty Dictionary: {}
                          Empty Dictionary: {}
     Number of items: 1
                          Number of items: (1)
     key: Name
                          key: Name
    Σ ⊞
```

< Prev

Reset

Next >



Sample Test Cases

```
CTP17092...
                                         Submit
         # Linear Search Implementation
   2
3
         # Read the array of integers from
   4
         arr = list(map(int, input().split()))
   5
   6
         # Read the key element to search
   7
         key = int(input())
   8
         # Perform linear search
   9
  10
         found = False
  11
       v for index, value in enumerate(arr):
       v if value == key:
  12
  13
         ----print(index)
  14
         found = True
         break
  15
  16
  17
         # If element is not found, print
         "Not found"
       v if not found:
  18
  19
         print("Not found")
  20
    Average time
                          Maximum time
                      <u>0.043</u> s
    0.041 s
                                             2 out of 2 shown test case(s) passed
    2 out of 2 hidden test case(s) passed
    Test case 1 (35 ms)
                           🏗 Debug
     Expected output
     1 2 3 4 3 5 6
                          1 2 3 4 3 5 6
     3
                          3
     2
```

< Prev

Reset

Next >

