

## Practical 2

### About this unit

Practical 2

#### Practice Lab Assignment

Unit • 100% completed



#### Lab Assignment

Unit • 100% completed



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



```

1  l = []
2  while True :
3      print("1. Add\n2. Remove\n3. Display\n4. Quit")
4      n = int(input("Enter choice: "))
5      if n == 1:
6          a = int(input("Integer: "))
7          l.append(a)
8          print(f"List after adding: {l}")
9      elif n == 2:
10         if (len(l) != 0):
11             b = int(input("Integer: "))
12             if b in l:
13                 l.remove(b)
14                 print(f"List after removing: {l}")
15             else:
16                 print("Element not found")
17         else:
18             print("List is empty")
19         elif n == 3:
20             if (len(l) == 0):
21                 print("List is empty")
22             else:
23                 print(l)
24         elif n==4:
25             break
26         else:
27             print("Invalid choice")

```

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



```
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
11 update_key = input("Enter the key to
update: ")
12 if update_key in dict:
13     new_value = input("Enter the new
value: ")
14     dict[update_key] = new_value
15     print("Value updated")
16 else:
17     print("Key not found")
18
19 retrieve_key = input("Enter the key
to retrieve: ")
20 if retrieve_key in dict:
21     print(f"Key: {retrieve_key},
Value: {dict[retrieve_key]}")
22 else:
23     print("Key not found")
24
25 get_key = input("Enter the key to
get using the get() method: ")
26 value = dict.get(get_key, "Key not
found")
27 if value != "Key not found":
28     print(f"Key: {get_key}, Value:
{value}")
29 else:
30     print(value)
31
32 deleted_key = input("Enter the key
to delete: ")
33 if deleted_key in dict:
34     del dict[deleted_key]
35     print("Deleted")
36 else:
37     print("Key not found")
38
39 print("Updated Dictionary:", dict)
```

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



```
1 arr = list(map(int,input().split("
2
3 key = int(input())
4
5 for i in range(len(arr)):
6     if arr[i] == key:
7         print(i)
8         break
9
10 if arr[i] != key:
11     print("Not found")
```

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



```
1 heights =  
  list(map(int,input().split(" ")))  
2  
3 captain = max(heights)  
4  
5 print(captain)
```