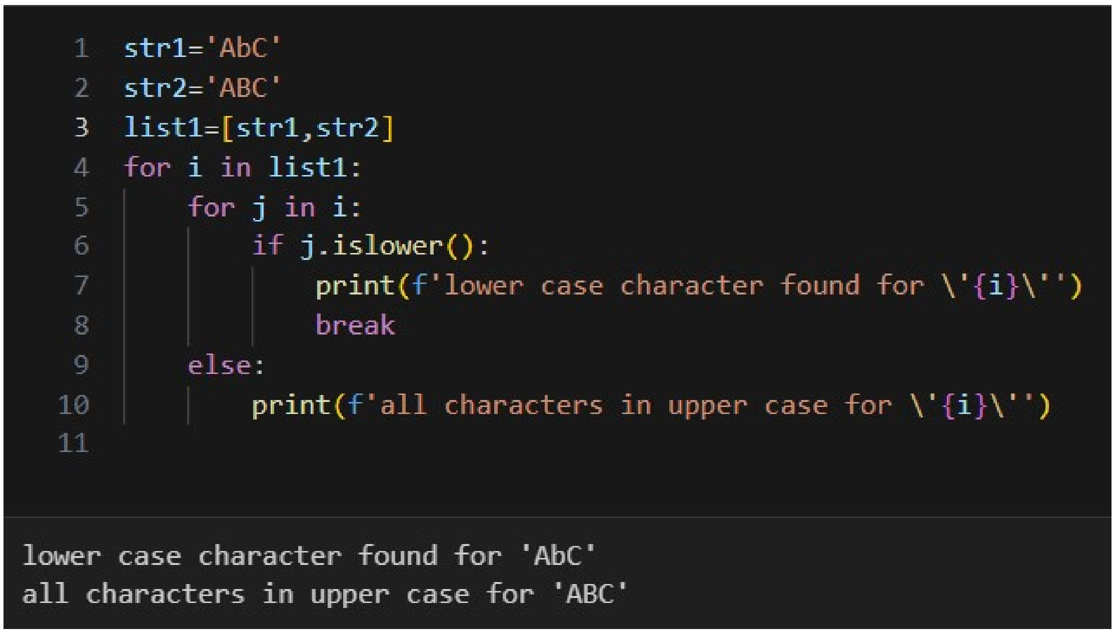
Assignment 2

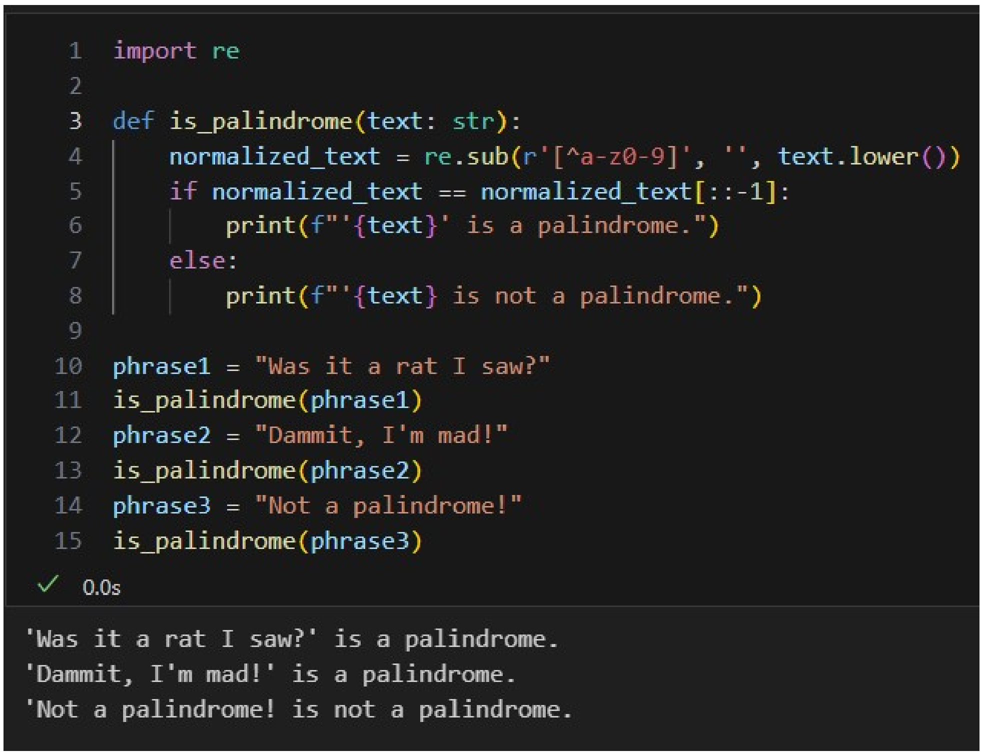
Q. 1. Check if all letters in a string are uppercase



```python
str1='AbC'
str2='ABC'
list1=[str1,str2]
for i in list1:
    for j in i:
        if j.islower():
            print(f'lower case character found for \'{i}\'')
            break
    else:
        print(f'all characters in upper case for \'{i}\'')

```

```
lower case character found for 'AbC'
all characters in upper case for 'ABC'
```

Q. 2. Write a version of a palindrome recognizer that also accepts phrase palindromes such as : Was it a rat I saw? or Dammit, I'm mad!Note that punctuation, capitalization, and spacing are usually ignored.

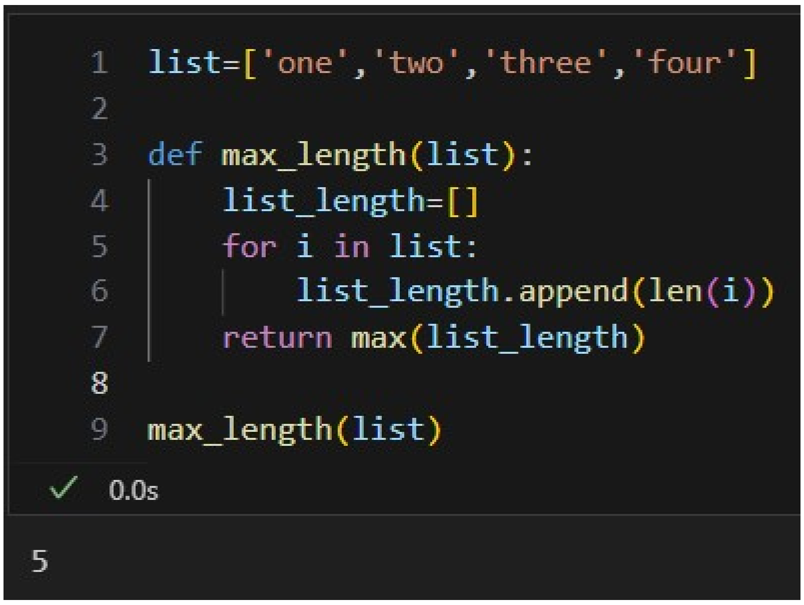


```python
import re

def is_palindrome(text: str):
    normalized_text = re.sub(r'[^a-z0-9]', '', text.lower())
    if normalized_text == normalized_text[::-1]:
        print(f"'{text}' is a palindrome.")
    else:
        print(f"'{text} is not a palindrome.")

phrase1 = "Was it a rat I saw?"
is_palindrome(phrase1)
phrase2 = "Dammit, I'm mad!"
is_palindrome(phrase2)
phrase3 = "Not a palindrome!"
is_palindrome(phrase3)
```

✓ 0.0s

```
'Was it a rat I saw?' is a palindrome.
'Dammit, I'm mad!' is a palindrome.
'Not a palindrome! is not a palindrome.
```

Q.3. Write a Python function that takes a list of words and returns the length of the longest one

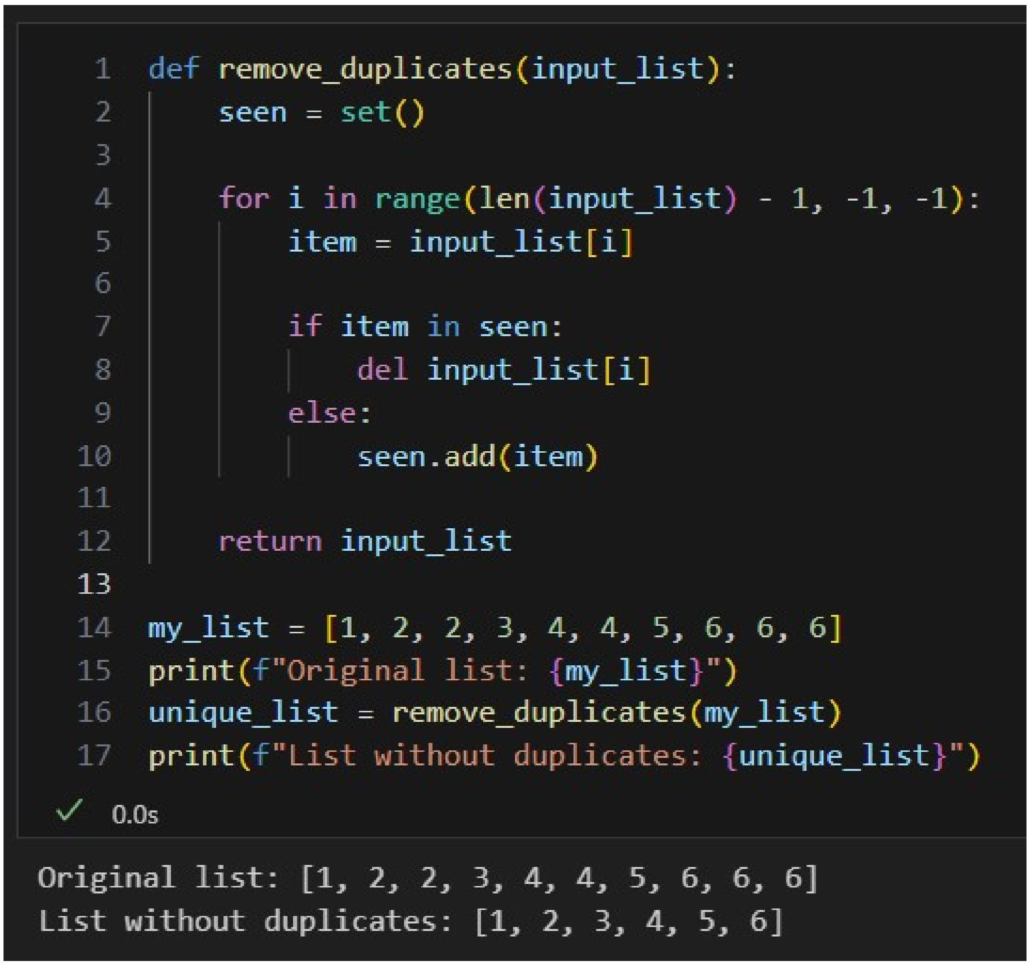


```
list=['one','two','three','four']

def max_length(list):
    list_length=[]
    for i in list:
        list_length.append(len(i))
    return max(list_length)

max_length(list)
```

✓ 0.0s

```
5
```

Q.4. Write a Python program to remove duplicates from a list



```
def remove_duplicates(input_list):
    seen = set()

    for i in range(len(input_list) - 1, -1, -1):
        item = input_list[i]

        if item in seen:
            del input_list[i]
        else:
            seen.add(item)

    return input_list

my_list = [1, 2, 2, 3, 4, 4, 5, 6, 6, 6]
print(f"Original list: {my_list}")
unique_list = remove_duplicates(my_list)
print(f"List without duplicates: {unique_list}")
```

✓ 0.0s

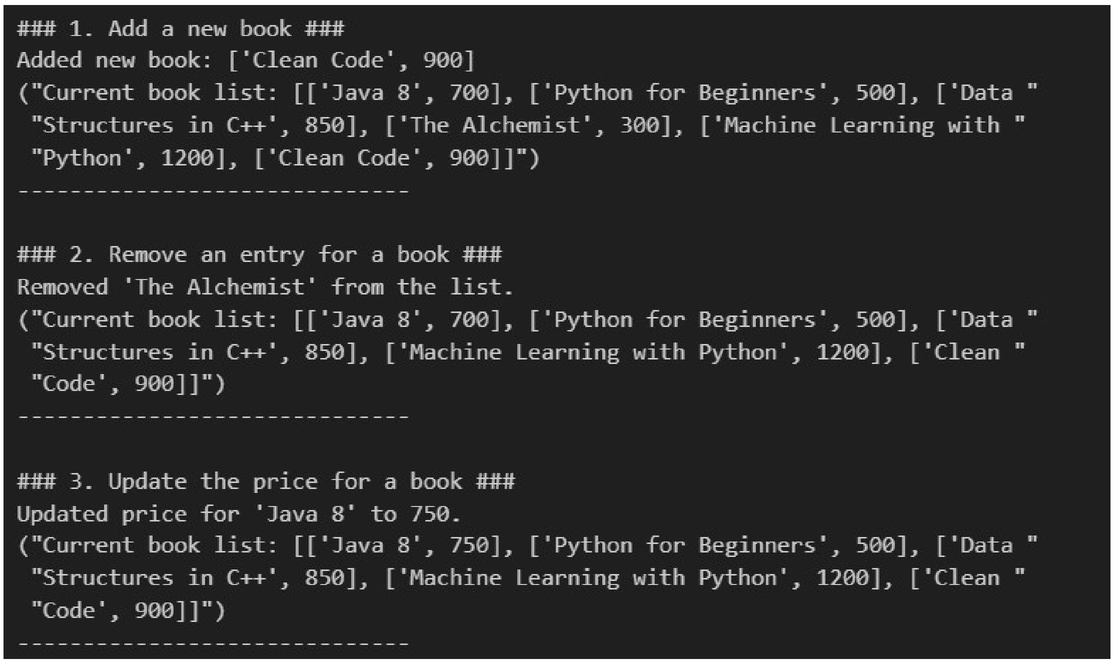
```
Original list: [1, 2, 2, 3, 4, 4, 5, 6, 6, 6]
List without duplicates: [1, 2, 3, 4, 5, 6]
```

Q 5. Create a list of books

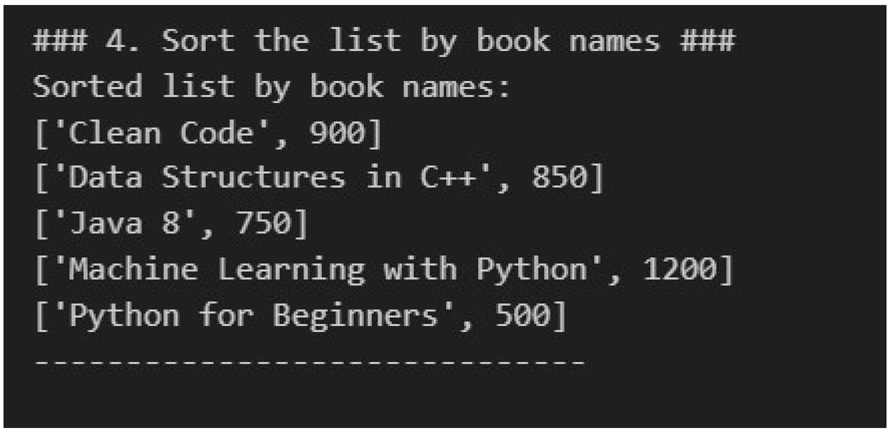
e.g. booklist =[['Java 8', 700], ['Python for Beginners', 500],....]

Perform following operations on the list

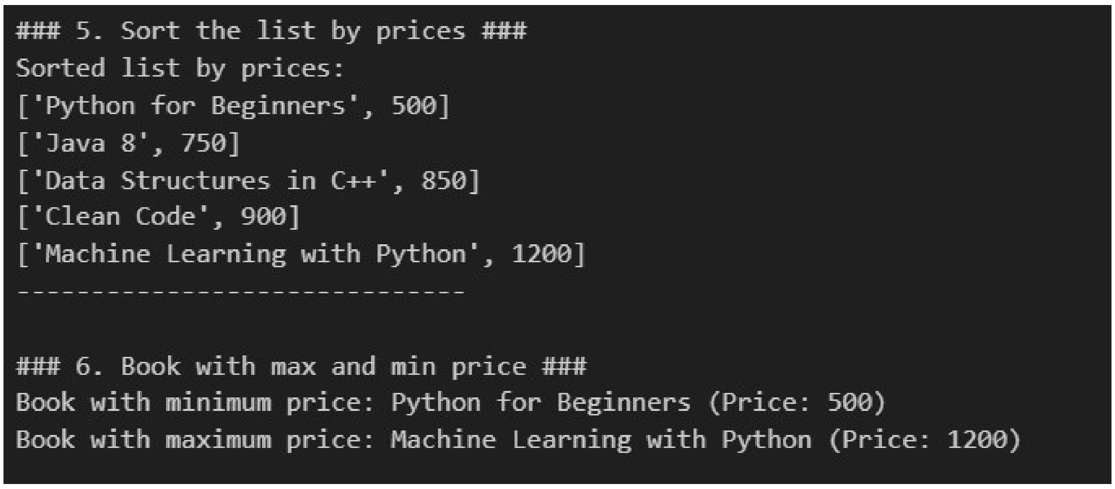
1. Add a new book with price
2. Remove entry for a book
3. update price for a book
4. Sort the list by book names
5. Sort the list by prices
6. Print the book with max and min price [hint : you may use min()/max() functions of python]



```
### 1. Add a new book ###
Added new book: ['Clean Code', 900]
("Current book list: [['Java 8', 700], ['Python for Beginners', 500], ['Data "
 "Structures in C++', 850], ['The Alchemist', 300], ['Machine Learning with "
 "Python', 1200], ['Clean Code', 900]]")
------------------------------

### 2. Remove an entry for a book ###
Removed 'The Alchemist' from the list.
("Current book list: [['Java 8', 700], ['Python for Beginners', 500], ['Data "
 "Structures in C++', 850], ['Machine Learning with Python', 1200], ['Clean "
 "Code', 900]]")
------------------------------

### 3. Update the price for a book ###
Updated price for 'Java 8' to 750.
("Current book list: [['Java 8', 750], ['Python for Beginners', 500], ['Data "
 "Structures in C++', 850], ['Machine Learning with Python', 1200], ['Clean "
 "Code', 900]]")
------------------------------
```



```
### 4. Sort the list by book names ###
Sorted list by book names:
['Clean Code', 900]
['Data Structures in C++', 850]
['Java 8', 750]
['Machine Learning with Python', 1200]
['Python for Beginners', 500]
------------------------------
```



```
### 5. Sort the list by prices ###
Sorted list by prices:
['Python for Beginners', 500]
['Java 8', 750]
['Data Structures in C++', 850]
['Clean Code', 900]
['Machine Learning with Python', 1200]
------------------------------

### 6. Book with max and min price ###
Book with minimum price: Python for Beginners (Price: 500)
Book with maximum price: Machine Learning with Python (Price: 1200)
```

Q.6. Write a Python program to compute element-wise sum of given tuples, using "zip()" function

Original tuples:

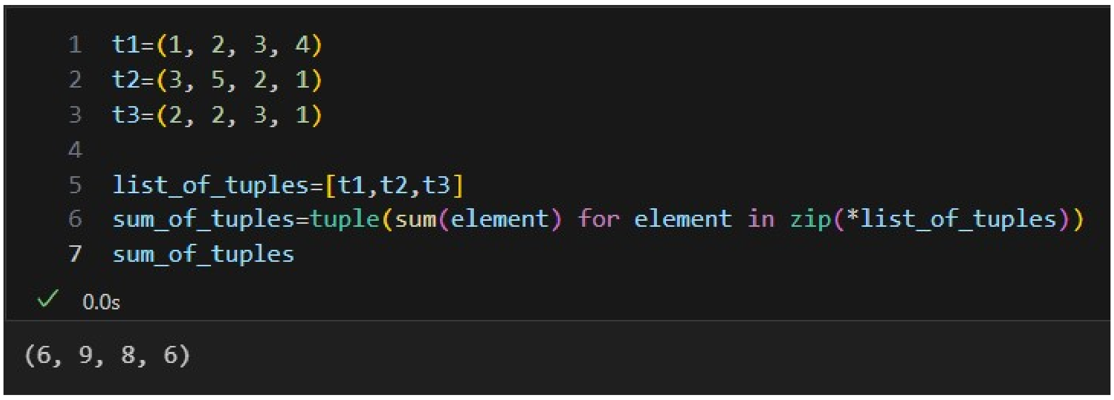
(1, 2, 3, 4)

(3, 5, 2, 1)

(2, 2, 3, 1)

Element-wise sum of the said tuples:

(6, 9, 8, 6)



```python
t1=(1, 2, 3, 4)
t2=(3, 5, 2, 1)
t3=(2, 2, 3, 1)

list_of_tuples=[t1,t2,t3]
sum_of_tuples=tuple(sum(element) for element in zip(*list_of_tuples))
sum_of_tuples
```

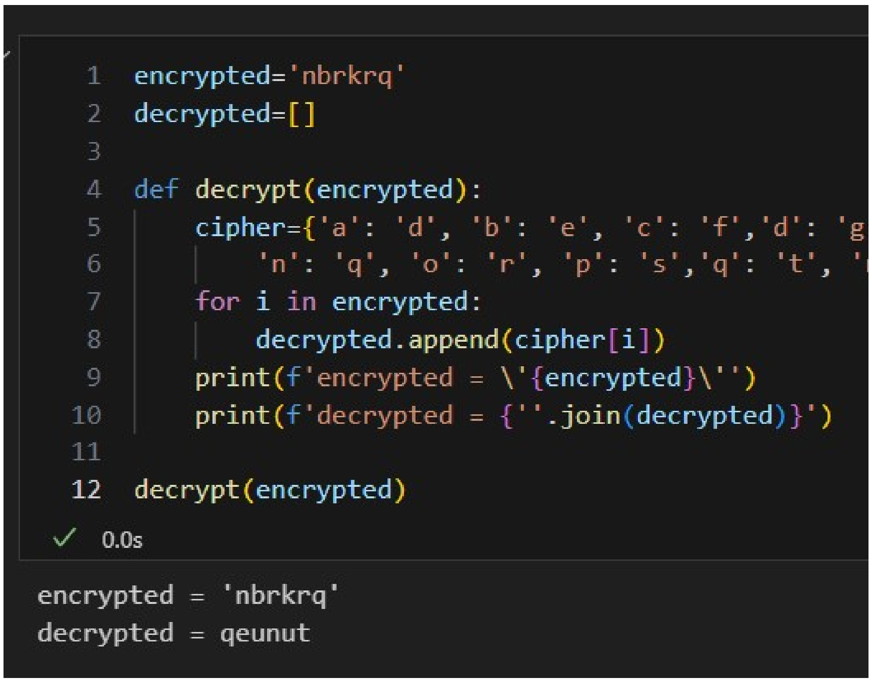
✓ 0.0s

```
(6, 9, 8, 6)
```

Q.7 In cryptography, a Caesar cipher is a very simple encryption techniques in which each letter in the plain text is replaced by a letter some fixed number of positions down the alphabet. For example, with a shift of 3, A would be replaced by D, B would become E, and so on. Create a cipher to represent each key with corresponding value as :

{'a': 'd', 'b': 'e', 'c': 'f', 'd': 'g', 'e': 'h', 'f': 'i', 'g': 'j', 'h': 'k', 'i': 'l', 'j': 'm', 'k': 'n', 'l': 'o', 'm': 'p', 'n': 'q', 'o': 'r', 'p': 's', 'q': 't', 'r': 'u', 's': 'v', 't': 'w', 'u': 'x', 'v': 'y', 'w': 'z', 'x': 'a', 'y': 'b', 'z': 'c'} encrypted = 'nbrkrq'

Expected output : decrypted = python



```python
encrypted='nbrkrq'
decrypted=[]

def decrypt(encrypted):
    cipher={'a': 'd', 'b': 'e', 'c': 'f','d': 'g
        'n': 'q', 'o': 'r', 'p': 's','q': 't', '
    for i in encrypted:
        decrypted.append(cipher[i])
    print(f'encrypted = \'{encrypted}\'')
    print(f'decrypted = {''.join(decrypted)}')

decrypt(encrypted)
```

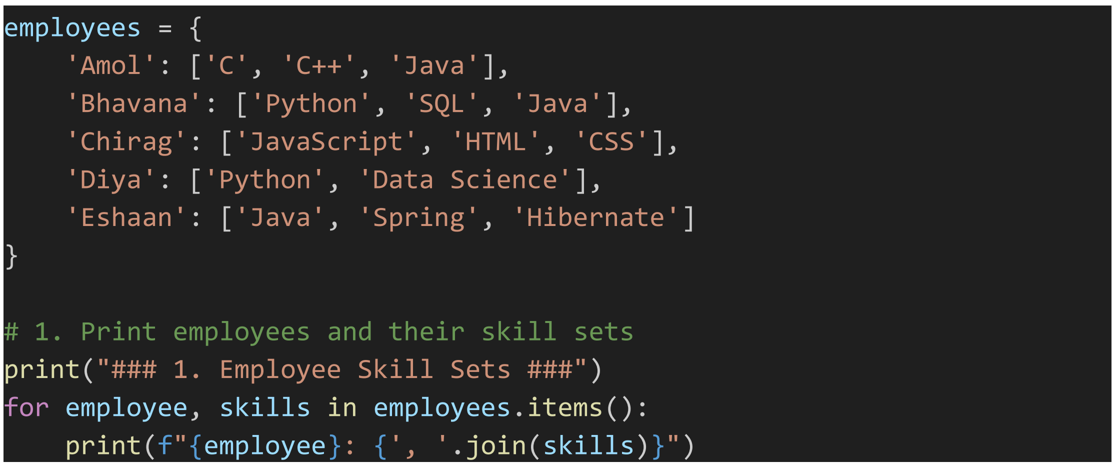
```
✓ 0.0s
```

```
encrypted = 'nbrkrq'
decrypted = qeunut
```

Q.8 For a given dictionary [Add few more entries]

employees = {'Amol' : ['C', 'C++','Java'],.....}

1. print employees and their skill sets
2. Find all the employees who know Java
3. Update skill for an employee
4. Add/remove employee data



```python
employees = {
    'Amol': ['C', 'C++', 'Java'],
    'Bhavana': ['Python', 'SQL', 'Java'],
    'Chirag': ['JavaScript', 'HTML', 'CSS'],
    'Diya': ['Python', 'Data Science'],
    'Eshaan': ['Java', 'Spring', 'Hibernate']
}

# 1. Print employees and their skill sets
print("### 1. Employee Skill Sets ###")
for employee, skills in employees.items():
    print(f"{employee}: {', '.join(skills)}")
print("-" * 30)
```

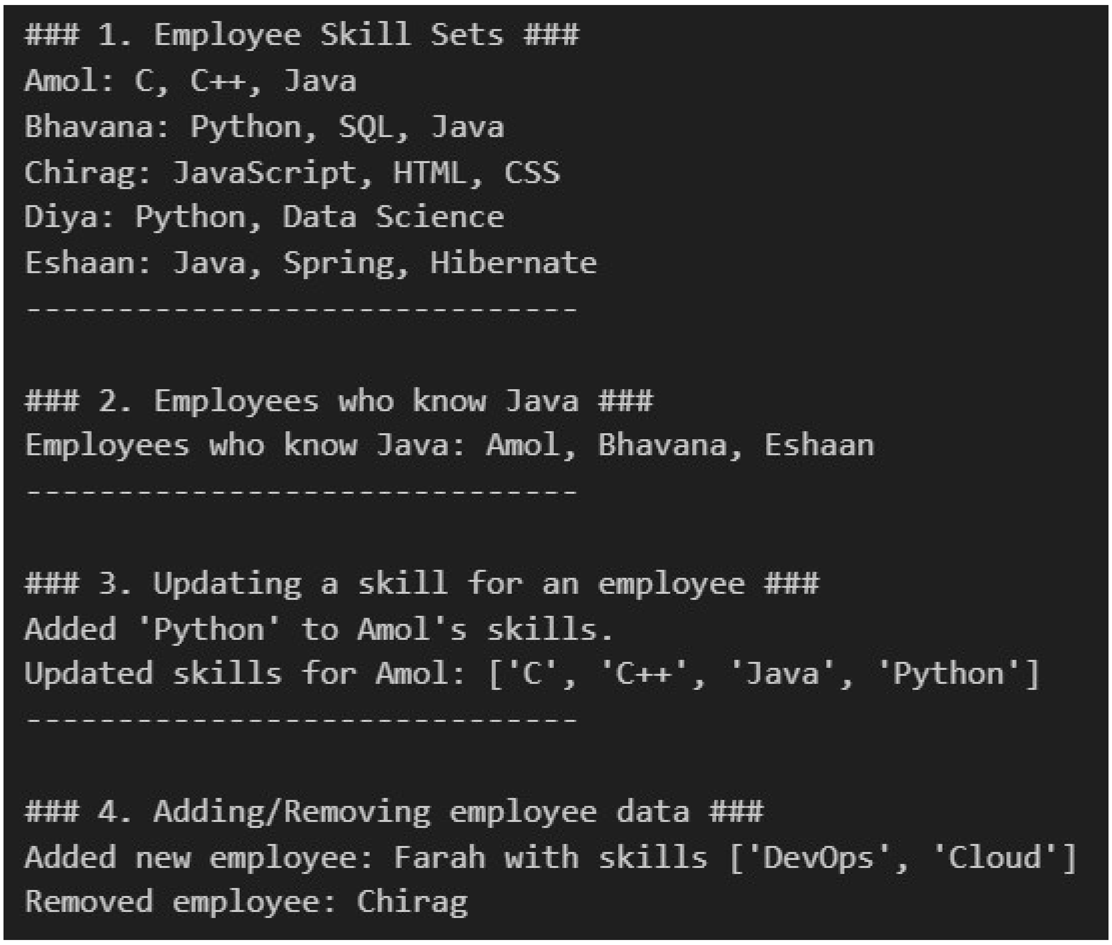
```python
# 2. Find all the employees who know Java
print("\n### 2. Employees who know Java ###")
java_employees = []
for employee, skills in employees.items():
    if 'Java' in skills:
        java_employees.append(employee)
print("Employees who know Java:", ', '.join(java_employees))
print("-" * 30)
```

```python
# 3. Update skill for an employee
print("\n### 3. Updating a skill for an employee ###")
employee_to_update = 'Amol'
new_skill = 'Python'
if employee_to_update in employees:
    if new_skill not in employees[employee_to_update]:
        employees[employee_to_update].append(new_skill)
        print(f"Added '{new_skill}' to {employee_to_update}'s skills.")
    else:
        print(f"{employee_to_update} already knows '{new_skill}'.")
print(f"Updated skills for {employee_to_update}:
{employees[employee_to_update]}")
print("-" * 30)
```

```python
# 4. Add/remove employee data
print("\n### 4. Adding/Removing employee data ###")
```

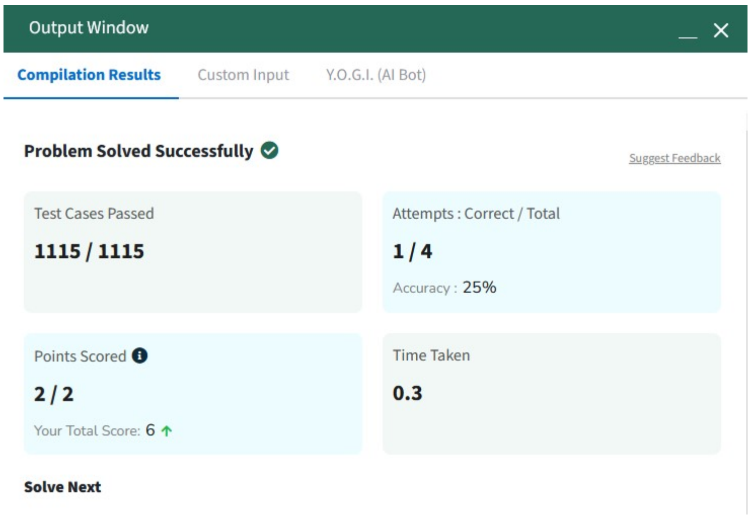
```python
# Add a new employee
new_employee = 'Farah'
new_skills = ['DevOps', 'Cloud']
employees[new_employee] = new_skills
print(f"Added new employee: {new_employee} with skills {new_skills}")
```

```python
# Remove an employee
employee_to_remove = 'Chirag'
if employee_to_remove in employees:
    del employees[employee_to_remove]
    print(f"Removed employee: {employee_to_remove}")
else:
    print(f"Employee {employee_to_remove} not found.")
```



```
### 1. Employee Skill Sets ###
Amol: C, C++, Java
Bhavana: Python, SQL, Java
Chirag: JavaScript, HTML, CSS
Diya: Python, Data Science
Eshaan: Java, Spring, Hibernate
------------------------------

### 2. Employees who know Java ###
Employees who know Java: Amol, Bhavana, Eshaan
------------------------------

### 3. Updating a skill for an employee ###
Added 'Python' to Amol's skills.
Updated skills for Amol: ['C', 'C++', 'Java', 'Python']
------------------------------

### 4. Adding/Removing employee data ###
Added new employee: Farah with skills ['DevOps', 'Cloud']
Removed employee: Chirag
```

Missing in Array


Output Window

Compilation Results   Custom Input   Y.O.G.I. (AI Bot)

Problem Solved Successfully   Suggest Feedback

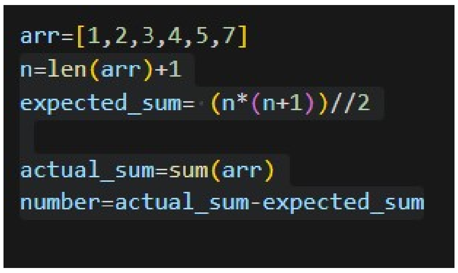
Test Cases Passed
1115 / 1115

Attempts : Correct / Total
1 / 4
Accuracy : 25%

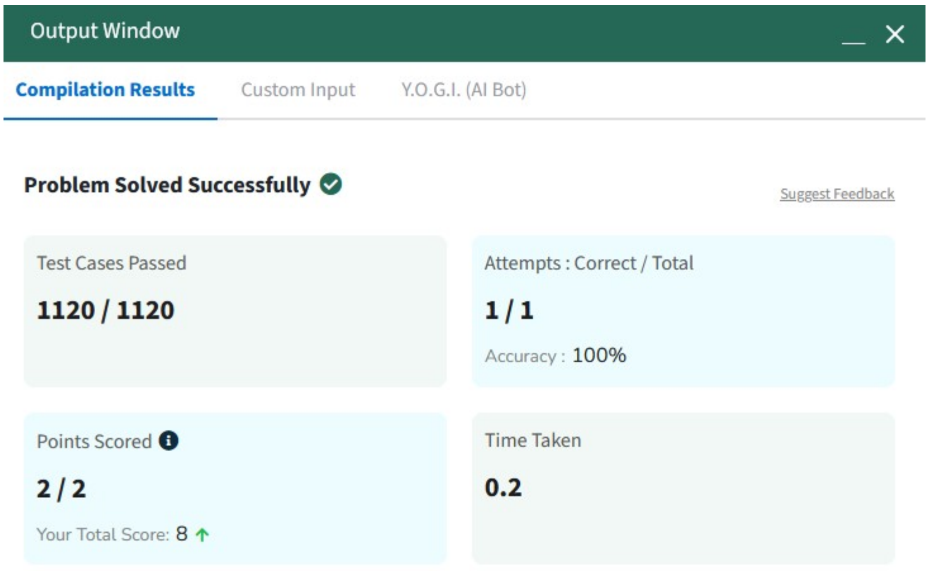
Points Scored
2 / 2
Your Total Score: 6

Time Taken
0.3

Solve Next


```
arr=[1,2,3,4,5,7]
n=len(arr)+1
expected_sum= (n*(n+1))//2

actual_sum=sum(arr)
number=actual_sum-expected_sum
```

Second Largest (GeekforGeeks)


Output Window

Compilation Results   Custom Input   Y.O.G.I. (AI Bot)

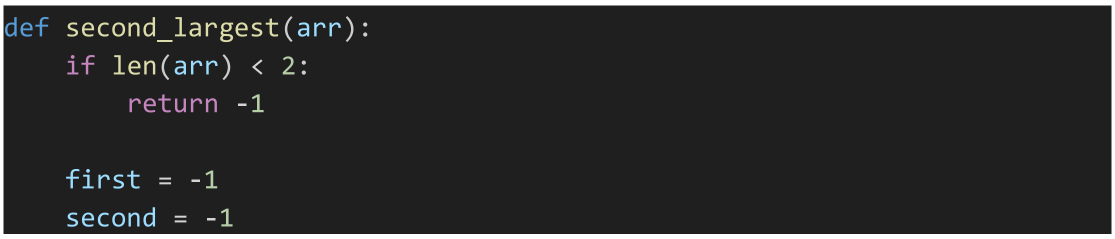
Problem Solved Successfully   Suggest Feedback

Test Cases Passed
1120 / 1120

Attempts : Correct / Total
1 / 1
Accuracy : 100%

Points Scored
2 / 2
Your Total Score: 8

Time Taken
0.2


```
def second_largest(arr):
    if len(arr) < 2:
        return -1

    first = -1
    second = -1
```

```python
    for num in arr:
        if num > first:
            second = first
            first = num
        elif num > second and num < first:
            second = num
```

```python
    if second == -1:
        return -1
    else:
        return second
second_largest([1,2,3,4,5,6])
```