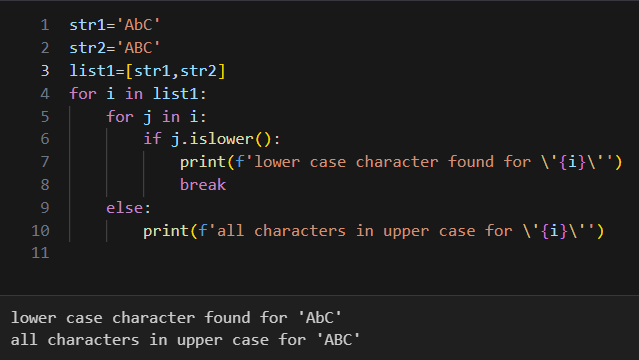
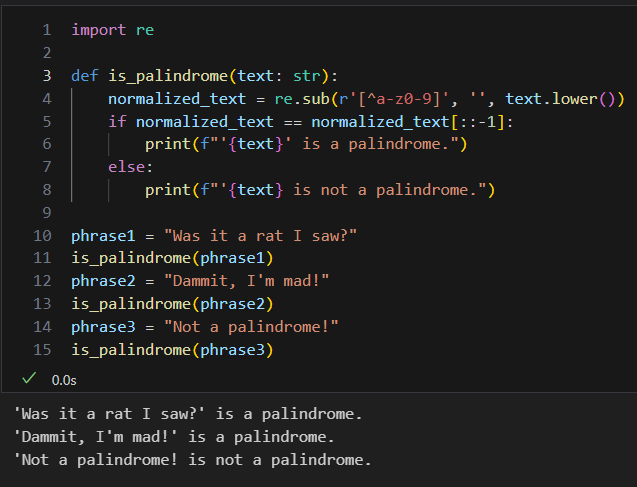
Assignment 2

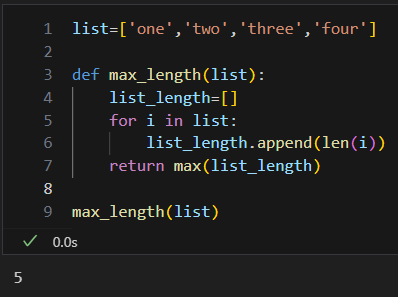
1. 1. Check if all letters in a string are uppercase



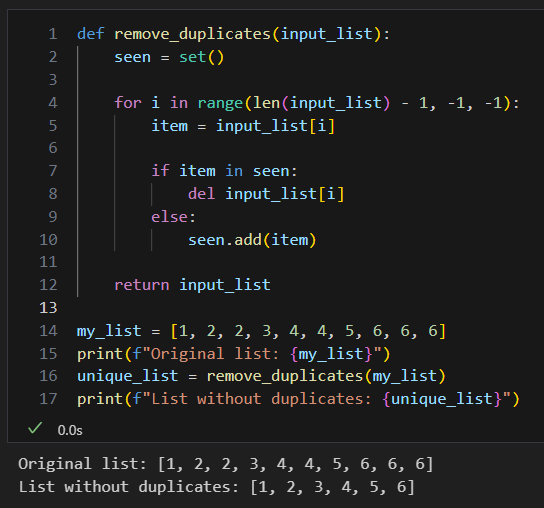
Q. 2. Write a version of a palindrome recognizer that also accepts phrase palindromes such as : Was it a rat I saw? or Dammit, I'm mad!Note that punctuation, capitalization, and spacing are usually ignored.



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



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



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

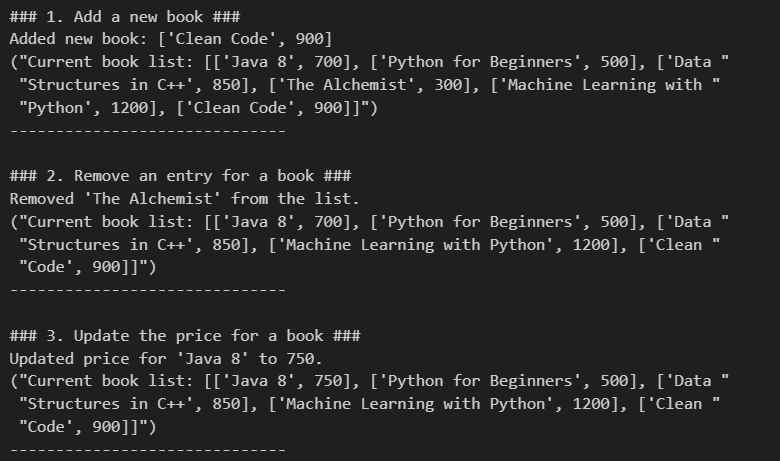
2. Remove entry for a book

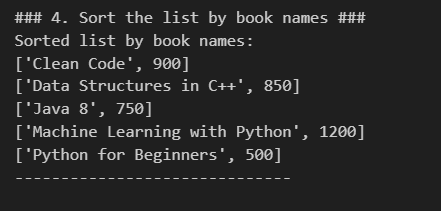
3. update price for a book

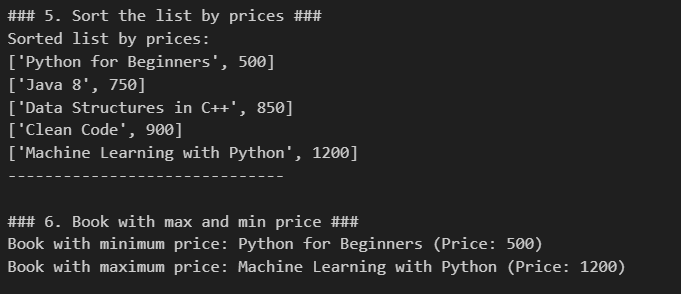
4. Sort the list by book names

5. Sort the list by prices

6. Print the book with max and min price [hint : you may use min()/max() functions of python]







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

Original tuples:

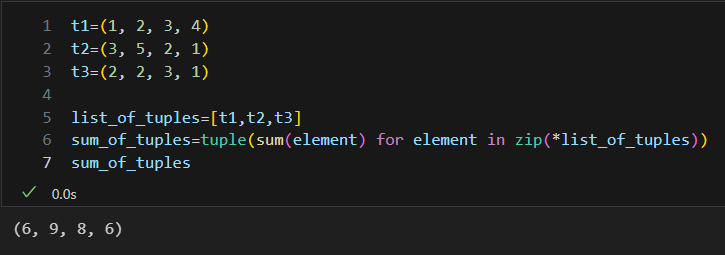
(1, 2, 3, 4)

(3, 5, 2, 1)

(2, 2, 3, 1)

Element-wise sum of the said tuples:

(6, 9, 8, 6)

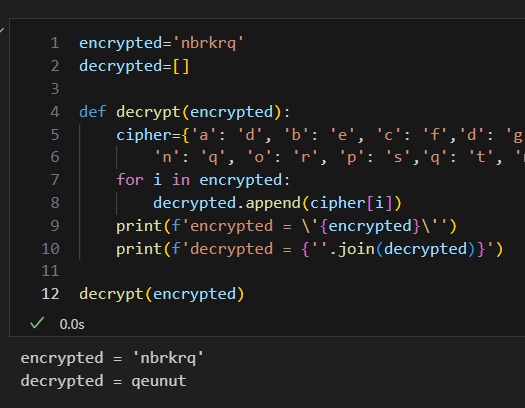


Q.7 In cryptography, a Caesar cipher is a very simple encryption techniques in which each letter in the plain text is replaced by a letter some fixed number of positions down the alphabet. For example, with a shift of 3, A would be replaced by D, B would become E, and so on. Create a cipher to represent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



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

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)

# 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)

# 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)

# 4. Add/remove employee data

print("\n### 4. Adding/Removing employee data ###")

# 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}")

# Remove an employee

employee\_to\_remove = 'Chirag'

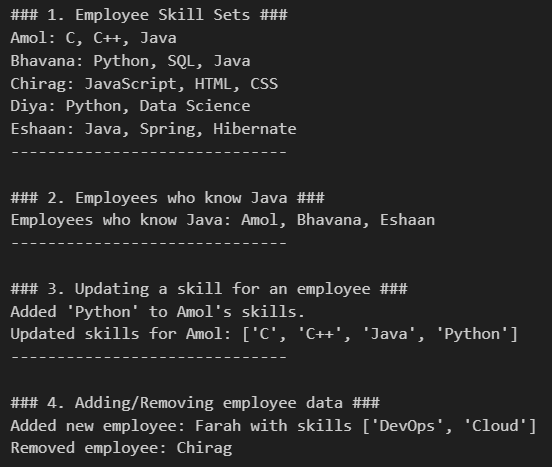
if employee\_to\_remove in employees:

    del employees[employee\_to\_remove]

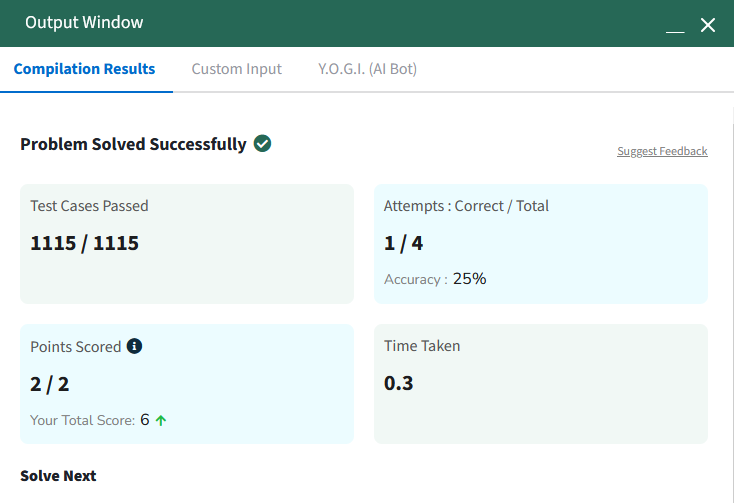
    print(f"Removed employee: {employee\_to\_remove}")

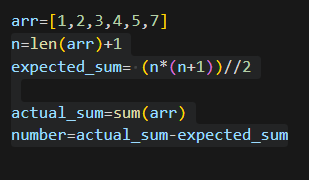
else:

    print(f"Employee {employee\_to\_remove} not found.")

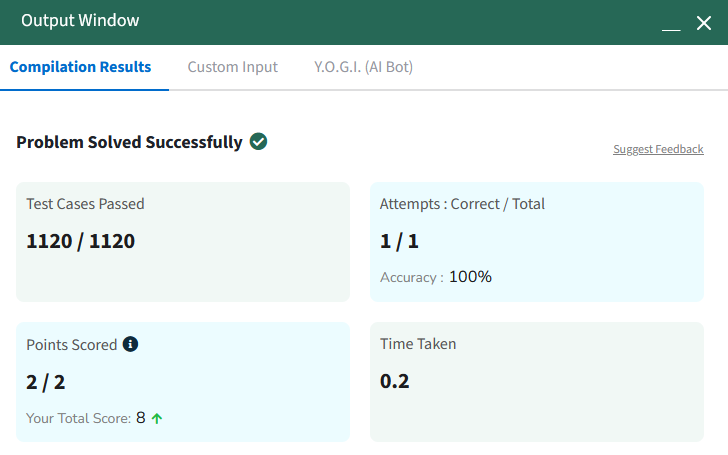


Missing in Array





Second Largest (GeekforGeeks)



def second\_largest(arr):

    if len(arr) < 2:

        return -1

    first = -1

    second = -1

    for num in arr:

        if num > first:

            second = first

            first = num

        elif num > second and num < first:

            second = num

    if second == -1:

        return -1

    else:

        return second

second\_largest([1,2,3,4,5,6])