ASSIGNMENT – I
ITO-804- Python Programming
Max Marks: 10
Submission Deadline: 10 April 2023

Questions	Course Outcome	Skill	Bloom's Level
Q1	**CO1:** Demonstrate thorough understanding of basic data types in Python and permissible operations on them.	**Applying**	3
Q2	**CO1:** Demonstrate thorough understanding of basic data types in Python and permissible operations on them.	**Applying**	3
Q3	**CO2:** Apply available programming constructs in Python to solve programming problems	**Applying**	3
Q4	**CO2:** Apply available programming constructs in Python to solve programming problems	**Applying**	3

Q1.) Write a Python program that takes a list of daily stock prices as input, and returns the best days to buy and sell stocks in order to maximize profit. The list contains the stock prices for each day, starting from the first day.

For example, the list (100, 180, 260, 310, 40, 535, 695) represents the stock prices for 7 days, where the price on the first day is 100, the second day is 180, and so on.

The program should find the best days to buy and sell stocks such that the profit obtained is maximum. For instance, in the given list of stock prices, the best days to buy and sell stocks would be:

Buy stock on the 1st day (price=100)
Sell stock on the 4th day (price=310)
 Buy stock on the 5th day (price=40)
Sell stock on the 7th day (price=695)
The program should output these buy and sell days as a tuple or list of integers. give me solution of this program.   **(2.5 Marks)**

Q2.) You are given a list of book titles and their corresponding publication years. Your task is to find the earliest year in which a trilogy of books was published. A trilogy is defined as a series of three books published in consecutive years.
For example, consider the following list of book titles and publication years:

titles = ['The Hunger Games', 'Catching Fire', 'Mockingjay', 'The Lord of the Rings', 'The Two Towers', 'The Return of the King', 'Divergent', 'Insurgent', 'Allegiant']
years = [2008, 2009, 2010, 1954, 1955, 1956, 2011, 2012, 2013]

The earliest year in which a trilogy was published is 1954.

Write a Python function **earliest_trilogy_year(titles: List[str], years: List[int]) -> Optional[int]** that takes two lists as input: **titles** containing the titles of the books, and **years** containing their corresponding publication years. The function should return the earliest year in which a trilogy of books was published, or **None** if no such trilogy exists.

Examples:

titles = ['Book1', 'Book2', 'Book3', 'Book4', 'Book5', 'Book6']
years = [2019, 2021, 2012, 2013, 2016, 2017]
print(earliest_trilogy_year(titles, years))

The earliest year in which a trilogy was published is : None

A trilogy is defined as a series of three books published in consecutive years.

Note:

- You can assume that the input lists are non-empty and contain an equal number of elements.
- If multiple trilogies exist with the same earliest year, return that year. **(2.5 Marks)**

Q3.) Write a Python program that reads in a CSV file of stock prices (e.g. ticker symbol, date, price), and then uses dictionaries and lists to calculate the highest and lowest prices for each stock from following table: **(2.5 Marks)**

Symbol	Date	Price
AAPL	2022-01-01	135.90
AAPL	2022-01-02	138.45
AAPL	2022-01-03	142.20
GOOG	2022-01-01	2105.75
GOOG	2022-01-02	2098.00
GOOG	2022-01-03	2125.50
MSFT	2022-01-01	345.20
MSFT	2022-01-02	344.70
MSFT	2022-01-03	342.10

Q4.) A) Write a Python program to remove duplicates from a list of lists. Sample list: [[10, 20], [40], [30, 56, 25], [10, 20], [33], [40]]

 B) Write a Python program which takes a list and returns a list with the elements "shifted left by one position" so [1, 2, 3] yields [2, 3, 1].

Example: [1, 2, 3] → [2, 3, 1]

[11, 12, 13] → [12, 13, 11]

C) Iterate a given list and count the occurrence of each element and create a dictionary to show the count of each element.

Original list [11, 45, 8, 11, 23, 45, 23, 45, 89, 11, 89]

And expected output is:

Printing count of each item   {11: 3, 45: 3, 8: 1, 23: 2, 89: 2}   **(2.5 Marks)**