**Course Title: Programming Language I**

**Course Code: CSE 110**

**Assignment no: 6**

| Coding tasks | 8 |
|---|---|
| Tracing | 3 |
| Total tasks | 11 |
| Total marks | 11 x 5 = 55 |

# Task 1

Write a Python program that reads 5 numbers from the user and enters it into a list.
Upon entering each number, show the list as output.
Complete the above task with and without the append function

| User Input (data type: int) | Output (data type: list) |
|:---:|:---:|
| 5 | [5] |
| 30 | [5, 30] |
| 2 | [5, 30, 2] |
| 4 | [5, 30, 2, 4] |
| -6 | [5, 30, 2, 4, -6] |

# Task 2

Take a list of 5 elements as user input and print the list in reverse order. Slicing can be used handling the given list as input. But to reverse the list you must not use slicing and split( ). *Hint: You may use a loop to show the reversed list as a new list.*

| User Input (data type: str) | Output (data type: list) |
|:---:|:---:|
| [5, -5, 100, 1, 0] | [0, 1, 100, -5, 5] |

# Task 3

Write a Python program that takes an input from the user, then prints a new list excluding the first and last two elements of the given list. In case of not enough elements in the list, print "Not possible". *Hint: May use list slicing but not split( )*

| User Input (data type: str) | Output (data type: list) |
|:---:|:---:|
| 10, 20, 24, 25, 26, 35, 70 | [24, 25, 26] |
| 10, 20, 24, 25 | [] |
| 10, 20, 24 | Not possible |

# Task 4

Let's assume, you have two given lists: List_one and List_two. Write a Python program that prints "True", if the given two lists have at least one common member. Otherwise, prints "False".
*Hint: Please use concepts of **flag** and **break** to solve this task.*

| Given list (data type: list) | Output |
|---|---|
| List_one : [1, 4, 3, 2, 6]<br>List_two : [5, 6, 9, 8, 7] | True |
| List_one : [1, 4, 3, 2, 5]<br>List_two : [8, 7, 6, 9] | False |

# Task 5

Suppose you have been given two lists. Write a Python program that replaces the last element of the first list with the second list. Note: Don't use remove() function.

| Given list (data type: list) | Output (data type: list) |
|---|---|
| List_one : [1, 4, 7, 5]<br>List_two : [6, 1, 3, 9] | [1, 4, 7, 6, 1, 3, 9] |
| List_one : [1, 3, 5, 7, 9, 10]<br>List_two : [2, 4, 6, 8] | [1, 3, 5, 7, 9, 2, 4, 6, 8] |

# Task 6

Suppose you have two given lists. Write a Python program that creates a new list with all the even elements of both of the given lists and prints the new list. Your program should work for any two given lists.

| Given list (data type: list) | Output (data type: list) |
|---|---|
| list1 = [1, 2, 3, 4, 5, 6, 7, 8, 9]<br>list2 = [10, 11, 12, -13, -14, -15,-16] | [2, 4, 6, 8, 10, 12, -14, -16] |

# Task 7

Design a recursive function called **nested_to_linear** that takes a nested list from the user, and returns the nested list as a linear one. Please refer to the example below to design your program and check with the sample input below.

Note: Recursion must be used in the nested_to_linear function instead of loop

| Function call |
|---|
| **nested_to_linear**(["start", 10, [4, 2, [11, [9, "mid", 3, [1, 0], 6]], 8], "Done"]) |
| **Output** |
| ['start', 10, 4, 2, 11, 9, 'mid', 3, 1, 0, 6, 8, 'Done'] |

# Task 8

Assume you have a class of five students. And as a faculty you need to enter their name and CSE110 GPA in a list called **grade_list**. This particular grade_list consists of individual lists of the student's name and GPA. Hence, it makes the grade_list a nested list / list of lists.

Particular student information will be given as input by the faculty. Please refer to the example below to design your program.

| Given list (data type: str) | Output (data type: list) |
|---|---|
| '[arif,3.7]'<br>'[syed,4.0]'<br>'[rafin,3.3]'<br>'[nuha,3.7]'<br>'[safi,3.0]' | [[arif,3.7]]<br>[[arif,3.7],[syed,4.0]]<br>[[arif,3.7],[syed,4.0],[rafin,3.3]]<br>[[arif,3.7],[syed,4.0],[rafin,3.3],[nuha,3.7]]<br>[[arif,3.7],[syed,4.0],[rafin,3.3],[nuha,3.7],[safi,3.0]] |

# Tracing 1

| | |
|---|---|
| 1 | `myList = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]` |
| 2 | `index1 = 0` |
| 3 | `index2 = 0` |
| 4 | `index1 = 1` |
| 5 | `while(index1<10):` |
| 6 | `    myList[index1] = index1+4` |
| 7 | `    index2 = 1` |
| 8 | `    while(index2<index1):` |
| 9 | `        myList[index1] = myList[index1] + myList[index2]-index1` |
| 10 | `        index2 = index2+1` |
| 11 | `    print(myList[index1])` |
| 12 | `    index1 = index1+1` |

| Output | |
|---|---|
| 1. 5 | 6. 77 |
| 2. 9 | 7. 143 |
| 3. 15 | 8. 273 |
| 4. 25 | 9. 531 |
| 5. 43 | 10. |

# Tracing 2

| | |
|---|---|
| 1 | `myList = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]` |
| 2 | `index1 = 0` |
| 3 | `index2 = 0` |
| 4 | `index1 = 1` |
| 5 | `while (index1 < 10):` |
| 6 | `    myList[index1] = index1 + 4` |
| 7 | `    index2 = 1` |
| 8 | `    while (index2 < index1):` |
| 9 | `        myList[index1] = myList[index1-1] - myList[index2-1] - index1` |
| 10 | `        index2 = index2 + 1` |
| 11 | `    print(myList[index1])` |
| 12 | `    index1 = index1 + 1` |

| Output | |
|---|---|
| 1. 5 | 6. -6 |
| 2. 3 | 7. -1 |
| 3. -5 | 8. -3 |
| 4. -12 | 9. -11 |
| 5. -12 | 10. |

# Tracing 3

| | |
|---|---|
| 1 | `myList = [10, 1, 4, 3, 8, 2, 4, 3, 5, 8, "done"]` |
| 2 | `index1, index2 = 0, 0` |
| 3 | `index1 = 1` |
| 4 | `while(index1<10):` |
| 5 | `  while(index2<index1):` |
| 6 | `    myList[index1] = myList[index1]+myList[index2]-index1` |
| 7 | `    index2 = index2+1` |
| 8 | `  print(myList[index1])` |
| 9 | `  index1 = index1+1` |
| 10 | `print(myList[-1:])` |

| Output | |
|---|---|
| 1. 10 | 6. 11 |
| 2. 12 | 7. 7 |
| 3. 12 | 8. 4 |
| 4. 16 | 9. 3 |
| 5. 13 | 10. ['done'] |