Course Title: Structural Programming Language
Course Code: CSE 1157
Assignment no: 5

This Assignment is to help you develop your concept of Arrays in Cpp.

[Please follow variable naming conventions including using meaningful variable names for all the tasks]
When you run your code, please make sure your outputs exactly match the sample outputs for each of the sample inputs given. Check if your code works for other valid inputs not given in the samples.

Write the code in Cpp to do the following tasks:

Task 1

Write a Cpp program that reads 5 numbers from the user into a array. After reading each number, print all the numbers that have been entered so far in the array.

Example:
If the user enters 3, prints “Numbers in the array: [3]”
If the user enters 5 next, prints “Numbers in the array: [3, 5]”
If the user enters 34, prints “Numbers in the array: [3, 5, 34]”
If the user then enters -11, prints “Numbers in the array: [3, 5, 34, -11]”
Finally, if the user enters 0 as the last number, prints “Numbers in the array: [3, 5, 34, -11, 0]”

Task 2

Write a Cpp program that takes an input from the user, then creates a new array excluding the first and last two elements of the given array and prints the new array. If there are not enough elements in the array to do the task, print "Not possible".

Note: You may use array slicing.

===

Sample Input 1:

```
10, 20, 24, 25, 26, 35, 70
```

Sample Output 1:

```
[24, 25, 26]
```

===

Sample Input 2:

```
10, 20, 24, 25, 26
```

Sample Output 2:

```
[24]
```

===

Sample Input 3:

```
10, 20, 24, 25
```

Sample Output 3:

```
[]
```

===

Sample Input 4:

```
10, 20, 24
```

Sample Output 4:

```
Not possible
```

Task 3

Write a Cpp program that reads 5 numbers from the user into a array, and then prints them in the reverse order.

Hint: You may create a array to store the input numbers and then use loop to print them in reverse order

===

Sample Input:

```
5
-5
100
1
0
```

Sample Output:

```
Printing values from the array in reverse order:
0
1
100
```

-5
5

Task 4

Write a Cpp program that turns every item of a array into its square. [Your program should work for any arrays; make changes to the arrays below and check whether your program works correctly]

===

Given array 1:
[1, 2, 3, 4, 5, 6, 7]

Sample Output 1:
[1, 4, 9, 16, 25, 36, 49]

===

Given array 2:
[3, 5, 1, 6]

Sample Output 2:
[9, 25, 1, 36]

Task 5

Write a Cpp program that removes all Empty strings from a **given array** and prints the modified array. [Your program should work for any given array; make changes to the array below and check whether your program works correctly]

Given Array:

["hey", "there", "", "what's", "", "up", "", "?"]

Sample Output:

Original Array: ['hey', 'there', '', "what's", '', 'up', '', '?']

Modified Array: ['hey', 'there', "what's", 'up', '?']

Task 6

Write a Cpp program that reads a string containing 7 numbers separated by commas, then makes a array of those numbers and prints the **largest number** and its location or index position in the array. [You are not allowed to use the max(), sort(), sorted() function here]

[Your program should work for any given array; make changes to the array below and check whether your program works correctly]

Hint: You may assume the first input to be the largest value initially and the largest value’s location to be 0.

Note: You may need to be careful while printing the output. Depending on your code, you might need data conversion.

===

Sample Input:
7, 13, 2, 10, 6, -11, 0

Sample Output:

```
My array: [7, 13, 2, 10, 6, -11, 0]
Largest number in the array is 13 which was found at index 1.
```

Task 7

Suppose you have been given two arrays. Write a Cpp program that replaces the last element of the first array with the second array. [Your program should work for any two given arrays; make changes to the arrays below and check whether your program works correctly]

==

Sample given arrays 1:
Array_one : [1, 4, 7, 5]
Array_two : [6, 1, 3, 9]

Sample Output 1:
[1, 4, 7, 6, 1, 3, 9]

==

Sample given arrays 2:
Array_one : [1, 3, 5, 7, 9, 10]
Array_two : [2, 4, 6, 8]

Sample Output 2:
[1, 3, 5, 7, 9, 2, 4, 6, 8]

Task 8

Assume, you have been given two arrays. [Your program should work for any two given arrays; change the following arrays and check whether your program works correctly for the code you have written]

array_one = [1, 2, 3, 4, 5, 6, 7, 8, 9]
array_two = [10, 11, 12, -13, -14, -15, -16]

Write a Cpp program that creates a new array with all the **even elements** of both of the given arrays and prints the new array.

Hint: You may create a third array to store the even elements of the given arrays.

==

Output for the above arrays: [2, 4, 6, 8, 10, 12, -14, -16]

Task 9

Write a Cpp program that reads a string as an input from the user where multiple numbers are separated by spaces. Then, make a array of numbers from the input string without using the split() function and print the array. Finally, remove all the occurrences of **even numbers** from the same input array and print the modified array.

===

Sample Input:

```
7 12 4 55 96 2 11 61 33 42
```

Sample Output:

```
Original array: [7, 12, 4, 55, 96, 2, 11, 61, 33, 42]
Modified array: [7, 55, 11, 61, 33]
```

Task 10

Write a Cpp program that reads a string as an input from the user where multiple numbers are separated by commas. Then, make a array of numbers from the input string and print the array. Finally, remove multiple occurrences of any numbers from the input array and print the modified array **without duplicate values**.

Hint: You may create a third array to store the results but you may try doing it in the same input array. You can use membership operators (in, not in) to make sure no duplicates are added.

==

Sample Input 1:

```
0, 0, 1, 2, 3, 4, 4, 5, 6, 6, 6, 7, 8, 9, 4, 4
```

Sample Output 1:

```
Input array: [0, 0, 1, 2, 3, 4, 4, 5, 6, 6, 6, 7, 8, 9, 4, 4]
Modified array: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

==

Sample Input 2:

```
7, 7, 7, 1, 0, 3, 3, 55, 9
```

Sample Output 2:

```
Input array: [7, 7, 7, 1, 0, 3, 3, 55, 9]
Modified array: [7, 1, 0, 3, 55, 9]
```

Task 11

Assume, you have been given two arrays: Array_one and Array_two. [Your program should work for any two given arrays; change the following arrays and check whether your program works correctly for the code you have written]

Write a Cpp program that prints "True", if the given two arrays have at least one common member. Otherwise, print "False".

Note: Please use concepts of flag and break to solve this task.

===

Given arrays 1:
Array_one : [1, 4, 3, 2, 6]
Array_two : [5, 6, 9, 8, 7]

Sample Output 1:
True

===

Given arrays 2:
Array_one : [1, 4, 3, 2, 5]
Array_two : [8, 7, 6, 9]

Sample Output 2:
False

Task 12

```
myArray = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
index1 = 0
index2 = 0
index1 = 1
while(index1<10):
  myArray[index1] = index1+4
  index2 = 1
  while(index2<index1):
      myArray[index1] = myArray[index1] + myArray[index2]-index1
      index2 = index2+1
  print(myArray[index1])
  index1 = index1+1
```

Output

Task 13

```
myArray = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
index1 = 0
index2 = 0
index1 = 1
while (index1 < 10):
  myArray[index1] = index1 + 4
  index2 = 1
  while (index2 < index1):
    myArray[index1] = myArray[index1-1] - myArray[index2-1] - index1
    index2 = index2 + 1
  print(myArray[index1])
  index1 = index1 + 1
```

Output

Task 14

```
myArray = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
b = []
index1 = 0
index2 = 0
index1 = 1
b = myArray
while(index1<10):
  myArray[index1] = index1+2
  index2 = 1
  while(index2<index1):
    myArray[index1] = b[index1]+myArray[index2]-index1
    index2 = index2+1
  print(myArray[index1])
  index1 = index1+1
```

Output

Task 15

```
myArray = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
b = []
index1 = 0
index2 = 0
index1 = 1
b = myArray
while (index1 < 10):
  myArray[index1] = index1 + 1
  index2 = 1
  while (index2 < index1):
    myArray[index1] = b[index2-1] + myArray[index2] - index1
    index2 = index2 + 1
  print(myArray[index1])
  index1 = index1 + 1
```

Output