## Practice 28 – Lab 13 - Assignment 1

This document is multi-purpose. You have to solve it for both Lab 12 and Assignment 1. You have to show the solutions in lab dated 4th May, 2023. Viva will be conducted and marks will be awarded in Lab as well as in assignment in theory course.

**Task 01:** Create a list/ array of 10 elements with random values of your choice. Print values in reverse order that is print array/ list from last element to first element

**Task 02:** Write a program to find the sum of all elements of the array

**Task 03:** Write a program to copy array/ list into another array/ list

**Task 04:** Take two arrays of five elements each, where elements are in ascending order. Create another array of 10 elements. Write code to merge previous two arrays into new array in ascending order

**Task 05:** Write a program to declare an array of ten elements. Initialize them randomly in any range. Print even and odd elements in separate rows. Count both elements. If there are more even elements, make all elements even by adding one or if there are more odd elements, make all elements odd by subtracting one. At the end print all elements in a single line.

**Task 06:** Write a program, declare an array of ten elements. Initialize them randomly with values 1 to 15, where value should not repeat.

**Task 07:** Write a program, declare an array of ten elements. Initialize them randomly with values 1 to 5. Check and print index of same element exists at different locations.

**Task 08:** Write a program, declare an array of 20 elements. Initialize elements randomly such that each element should be larger than previous element. Print the missing values in the list. See sample run carefully:

```
1 3 4 6 7 8 9 13 15 18 20 22 24 26 33 35 37 38 42 45
```

Output:

```
2 5 10 11 14 16 18 19 21 23 27 28 29 30 31 32 34 36 39 40 41 43 44
```

**Task 09:** Write a program, declare an array of 20 elements. Initialize elements at random in range 0-9. Next, average out each element with value of 4 neighbors. Two neighbors from left side and two from right side. Leave first two values and last two values, see explanation to get understanding:

**Explanation:**

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

First line has 20 random values in range 0-9. In second row, first two and last two values are same. The third value is average of first, second, fourth and fifth value. The sum is (3+2+1+2) = 8 / 4 = 2. Consider next value. The sum is (2 + 0 + 1 +2) = 5 / 4 = 1.

Each time we will consider updated values, means we will use same array / list for calculation and modification. Next, sum is (2 + 1 + 4 + 6) = 13 / 4 = 3. Similarly, consider last value. The sum is (1 + 1 + 3 + 4) = 9 / 2 = 2

**Task 10:** Initialize an array of 10 elements with random values in range 3-7. For each element print stars in single line. Consider sample run for understanding:

**Sample Run:**

```
3 6 4 1 ...
***
******
****
*
```

**Task 11:** Initialize an array of 10 elements with random values in range 3-7. Print elements of array in triangular style:

**Sample Run:**

```
3 6 4 1 0 2 7 1 2 3
3
3 6
3 6 4
3 6 4 1
3 6 4 1 0
...
```

**Task 12:** Modify task 11, print sum of elements with elements:

**Sample Run:**

```
3 6 4 1 0 2 7 1 2 3
3 = 3
3 6 = 9
3 6 4 = 13
3 6 4 1 = 14
3 6 4 1 0 = 14
...
```

**Task 13:** Modify task 11 and print elements in triples. There will be 8 triples for ten elements:

**Sample Run:**

```
3 6 4 1 0 2 7 1 2 3
3 6 4
6 4 1
4 1 0
1 0 2
...
```

**Task 14:** Create a 2D list of size 10 x 10. Initialize value at random in range 0 - 9. Print values in tabular form. Next, print principal diagonals only in diagonal form:

**Sample Run:**

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

```
  8
     2
        8
           9
              8
                 8
                    7
                       2
                          7
                             4
```

**Task 14:** Modify task 13. Run nested loop and replace zeros with ones and print values again:

**Sample Run:**

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

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

**Task 15:** Modify task 13 (not 14). Print indexes of zero values:

**Sample Run:**

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

4 7

. . .