

NATIONAL UNIVERSITY OF COMPUTER & EMERGING SCIENCES  
ISLAMABAD

**Programming Fundamentals (CS118)**

**Spring 2021 ASSIGNMENT # 6**

**Due Date: 31<sup>th</sup> May 2021**

**General Rubrics**

1. Mechanism for input validation
2. Use of proper indentation
3. Use of comments with code snippets
4. Meaningful variable and function names preferably using camelCase notation
5. Menu implementation using switch statements
6. Use of prompts before taking inputs from the user
7. Use of functions for independent subtasks in lengthy problems
8. Use of best suitable data types.
9. Use integer arrays unless specified
10. Use of nested braces for initialization of two dimensional arrays where each set of inner braces represents one row.

Total marks(10+10+10+10+10+35)= 85 marks

- 1) Write a C++ Program to Multiply Two Matrix Using Multi-dimensional Arrays. You can use int or double 2D array and takes it input from user. This program takes two matrices of order  $r1 \times c1$  and  $r2 \times c2$  respectively. Then, the program multiplies these two matrices (if possible) and displays it on the screen, please take input from user.
- 2) Take a sorted array of 15 elements of type int and a value x from user, you need to find the ceiling and floor of x. The ceiling of x is the smallest element in array greater than or equal to x, and the floor is the greatest element smaller than or equal to x. Assume that the array is sorted in ascending order. Write functions to find floor and ceiling of x from the given array.

**Examples :**

```

For example, let the input array be {1, 2, 8, 10, 10, 12, 19}
For x = 0:    floor doesn't exist in array, ceil = 1
For x = 1:    floor  = 1,   ceil  = 1
For x = 5:    floor  = 2,   ceil  = 8
For x = 20:   floor  = 19,  ceil doesn't exist in array

```

- 3) Take a Boolean 2D array of 10X 10, where each row is sorted from the user, Find the row with the maximum number of 1s.

**Example:**

```

Input matrix
0 1 1 1
0 0 1 1
1 1 1 1 // this row has maximum 1s
0 0 0 0

```

Output: 2

- 4) We are given two sorted array of size 15 and of type int (take input from user) (all elements are in ascending order. We need to merge these two arrays such that the initial numbers (after complete sorting) are in the first array and the remaining numbers are in the second array.
- 5) Take an unsorted array of size 15 and type int from the user and a number n, find if there exists a pair of elements in the array whose difference is n.

**Examples:**

```

Input: arr[] = {5, 20, 3, 2, 50, 80}, n = 78
Output: Pair Found: (2, 80)

```

```

Input: arr[] = {90, 70, 20, 80, 50}, n = 45
Output: No Such Pair

```

- 6) Write a Menu Driven C++ program that creates one-dimensional array arr[] and initialize it with user of size 15. The program should do following Tasks using Menu, The menu operations are implemented using functions: (7X5=35 marks)
- Write a function **count()**, that counts the occurrences of x (a number) in arr[].
  - Write a function **partition ()**, that take the first element of the array x and put x in a position such that all smaller elements (smaller than x) are before x, and put all greater elements (greater than x) after x.
  - Write a function **next\_XOR()**, the count of elements which are equal to the XOR of the next two elements in an array.

- d) Write a function `duplicates()`, which calculated the frequencies of all the elements and display them.
- e) Write a function **`circular()`**, which replace every element of the array by the sum of next two consecutive elements in a **circular** manner i.e.  $\text{arr}[0] = \text{arr}[1] + \text{arr}[2]$ ,  $\text{arr}[1] = \text{arr}[2] + \text{arr}[3]$ , ...  $\text{arr}[n - 1] = \text{arr}[0] + \text{arr}[1]$ .
- f) Write a function **`Search()`**, takes an array and element to search in the array and returns the index of element if the element is found. And return the negative number if not found.
- g) Write a function **`shift_circular ()`**, which shifts an array circularly left by two positions. Thus, if  $p[0] = 15$ ,  $p[1] = 30$ ,  $p[2] = 28$ ,  $p[3] = 19$  and  $p[4] = 61$  then after the shift  $p[0] = 28$ ,  $p[1] = 19$ ,  $p[2] = 61$ ,  $p[3] = 15$  and  $p[4] = 30$ .