National University of Computer and Emerging Sciences

Laboratory Manual

for

Programming Fundamentals

Course Instructor	Mr Waqas Mansoor
Lab Instructor(s)	Ms. Shazia Ahmed Mr. Adeel Qayyum
Section	PF E
Semester	Fall 2020

Department of Computer Science

FAST-NU, Lahore, Pakistan

Lab Manual 08

Hint: Take inputs in main function, pass as parameters to functions and print result return by functions in main. Use arrays wherever required

Problem 1:

Write a program to calculate students' average test scores and their grades. You may assume the following input data:

Johnson 85 83 77 91 76
Aniston 80 90 95 93 48
Cooper 78 81 11 90 73
Gupta 92 83 30 69 87
Blair 23 45 96 38 59
Clark 60 85 45 39 67
Kennedy 77 31 52 74 83
Bronson 93 94 89 77 97
Sunny 79 85 28 93 82
Smith 85 72 49 75 63

Use three arrays: a Cstring array to store the students' names, a (parallel) two-dimensional array to store the test scores, and a parallel one dimensional array to store grades. Your program must contain at least the following functions: a function to read and store data into two arrays, a function to calculate the average test score and grade, and a function to output the results. Have your program also output the class average.

Problem 2:

Write a user defined function named Upper-half() which takes a two dimensional array A, with size N rows and N columns as argument and prints the upper half of the array.

e.g.,

```
2 3 1 5 0                    2 3 1 5 0
7 1 5 3 1                      1 5 3 1
2 5 7 8 1   Output will be:      1 7 8
0 1 5 0 1                        0 1
3 4 9 1 5                          5
```

Problem 3:

Write a function in C++ which accepts a 2D array of integers and its size as arguments and displays the elements of middle row and the elements of middle column.
[Assuming the 2D Array to be a square matrix with odd dimension i.e. 3x3, 5x5, 7x7 etc...]

Sample Run:

Input: 2 dimension 3X3 array

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

Output through the function should be:

Middle Row: 7 6 9

Middle column: 5
6
1

Problem 4:

Write a menu driven C++ program to do following operation on two-dimensional array A of size m x n. You should use user-defined functions which accept 2-D array A, and its size m and n as arguments. The options are:

- To input elements into matrix of size m x n
- To display elements of matrix of size m x n
- Sum of all elements of matrix of size m x n
- To display row-wise sum of matrix of size m x n
- To display column-wise sum of matrix of size m x n
- To create transpose of matrix B of size n x m

Problem 5:

Write a program that randomly generates a 20 x 20 two-dimensional array, board, of type int. An element board[i][j] is a peak (either a maximum or a minimum) if all its neighbors (there should be either 3, 5, or 8 neighbors for any cell) are less than board[i][j], or greater than board[i][j]. The program should output all elements in board, with their indices, which are peak. It should also output if a peak is a maximum or a minimum.

Problem 6:

Write a program that simulates the rolling of two dice. The program should use rand to roll the first die and should use rand again to roll the second die. The sum of the two values should then be calculated. [*Note:* Each die can show an integer value from 1 to 6, so the sum of the two values will vary from 2 to 12, with 7 being the most frequent sum and 2 and 12 being the least frequent sums.] Your program should roll the two dice 36,000 times. Use a one-dimensional array to tally the numbers of times each possible sum appears. Print the results in a tabular format. Also, determine if the totals are reasonable (i.e., there are six ways to roll a 7, so approximately one-sixth of all the rolls should be 7).

Figure below shows the 36 possible combinations of the two dice.

	1	2	3	4	5	6
1	2	3	4	5	6	7
2	3	4	5	6	7	8
3	4	5	6	7	8	9
4	5	6	7	8	9	10
5	6	7	8	9	10	11
6	7	8	9	10	11	12

Sample Run:

Please enter seed: 2

Sum	Number of Times
2	95
3	243
4	276
5	402
6	496
7	609
8	459
9	378
10	318
11	211
12	113
	3600

Problem 7: (bonus)

The Tic-Tac-Toe Problem

Write a C++ program that allows two players to play tic-tac-toe.
Program Requirements:

1. Use an array to represent the playing board
2. Initialize the array to represent an empty board
3. Display the game board prior to every move and at the end of the game
4. prompt each player to move in their turn
    a. The player moves by entering a row and column number and the program marks the corresponding array element with either an X or an O
    b. The program validates each move by insuring that the row and column are inbounds and that the selected space is empty - if the player enters an invalid move, the program loops until a valid move is entered
    c. As a special case, if the player enters -1 for either the row or the column, the game ends
5. After each move, test to see if there is a winner
    a. If there is a winner, the program displays the board, announces the winner, and ends
6. Declare a draw and end when no moves remain

What is the Tic-Tac-Toe Game?

Tic-tac-toe is a game where two players X and O fill the hash (#) shaped box **(consist of two vertical lines crossing two horizontal lines)** with their alternate turns. The player who first fills the box with 3Xs or 3Os in a horizontal, vertical, or diagonal manner will win the game.

In some cases, when none of the players succeeds in filling the boxes horizontally, vertically, or diagonally with **3Xs or 3Os,** then the game will be considered to be a draw.