**Lab Tasks**

**Q1.** Write a recursive function to print the fibonacci sequence upto **n terms**. For example, if n = 6, your output should be: 1 1 2 3 5 8.

**Q2.** Write a function to swap the values of 2 variables passed **by reference**. Function definition will be as

follows: void swap(int\* var1, int\* var2) Print both of the original values in main(), then call the function

and print both the values in main() once again to show the values have been swapped.

**Q3.** Write a function with the declaration: **void manual\_strcpy(char\* str1, char\* str2, int len)** which takes these 2 strings as arguments and copies the contents of str2 into str1 **MANUALLY** (You CANNOT use any string function, NOT EVEN strlen). Initialise 2 char arrays in main(), print both the strings in main, call the function and then print both the strings in main once again to show the result. Make sure you null terminate the string after copying, otherwise you will get undesired output.

**Q4.** Declare a 2D array of size 10x5, and randomly generate values for each element of the array. Then, loop through the 2D array using a **POINTER** and print each element by dereferencing the pointer. Hint: Using syntax like arr[i][j] is not allowed. You must dereference the pointer manually by using \* operator.

**Q5.** Assume you have a 1D array of size N, with integers in sorted order and without any duplicates. You will simply pass this value of N to a function. Your function should use **recursion** to apply the following algorithm:

* Starting from left to right, remove the first number and every other number afterward until you reach the end of the list.
* Repeat the previous step again, but this time from right to left, remove the rightmost number and every other number from the remaining numbers.
* Keep repeating the steps again, alternating left to right and right to left, until a single number remains.

Your function should take **ONLY** the value of n (length of the array) as parameter. After applying the algorithm, your function should print the **last remaining number** in the array. **Hint: You don’t have to apply any operation on the array, you can solve it by simply using the value of N. Solved example:**

**Input:** n = 9

**Output:** 6

**Explanation:**

arr = [1, 2, 3, 4, 5, 6, 7, 8, 9]

arr = [2, 4, 6, 8]

arr = [2, 6]

arr = [6]

**Q6.** Tic tac toe is played on a 3x3 grid. The players alternate turns until 1 player gets 3 consecutive cells either vertically, horizontally, or diagonally. Your task is to create a tic tac toe AI bot to perform the best move so that the user can never win. To complete this, you will use the power of recursion.

Using recursion, you can check the **future moves** of both players to see whether the **current move** will end up being a good move or a bad move. Note that there are only 9 cells on the 3x3 grid, so there will be no more than 9 moves in any 1 game.