Programming Fundamentals COURSE & LAB

(BS-IT-F22 Morning)

Practice file 01

Assigned on: Saturday, December 02, 2023

Instructions:

- This is an individual practice file. Absolutely NO collaboration is allowed.
- Do NOT copy even a single line of code from any other person or book or Internet or any other source.
- This practice file has some bonus marks in your pf course.

Note:

There are two parts in this practice file. Questions in **Part A are related to the 1D array**. Questions in **Part B are related to 2D arrays.** We will assume in quizzes and in upcoming labs that you have solved all of these questions.

You may ask anything if you are stuck at any point while solving the following questions.

Solve these problems to enhance your logical skills with arrays.

PART A

Question # 1

Define a C function having two parameters and return the most frequent. This function should take an array of integers and its size as its first two parameters. This function should determine the most frequent element of the array, and it should return the **most frequent element** (i.e. the element which occurs the most often). In your function, you are NOT allowed to declare or use any other array.

Illustrate the working of your function (on various inputs) in a complete C program. See the following examples:

Example #	Elements of the array	Most frequent element	Frequency
1	**5 3 4 5 2 7 5 4**	**5**	**3**
2	**12 74 23 35 23 12**	**12** or **23**	**2**
3	**7 5 6**	**7** or **5** or **6**	**1**

As you can see from the last two examples, if more than one element is the most frequent element, then your function can determine any one of them as the most frequent element.

Illustrate the working of your function (on at least 5 different examples/inputs) in a complete C program.

Question # 2

Define a C function that removes all negative values from an unsorted array of integers. The function should preserve the order of remaining elements in the array. After removing negative values, the function should call the printArray(int arr[], int size) function to print the modified array, and it should return the count of negative values that were removed from the array. The function prototype is as follows:

int removeNegatives(int arr[ ], int size);

For example, if the array arr contains 7 elements { 11, -15, -2, 7, 11, 6, -8 }, then after the function call to removeNegatives, the array arr should now contain { 11, 7, 11, 6 } in its first four indices. The variable size of the array after removing negatives becomes 4 (since the partially filled array arr now contains 4 valid elements), and the function should return 3 (because 3 negative values were removed from the array).

Important Note: In the implementation of the function, you are not allowed to declare or use any other array. You are allowed to use the same array but potentially with a smaller size.

Question # 3

Define a C function:

void cyclicRotate (int arr [ ], int n, int k);

This function takes an array (arr) and its size (n) as its first two parameters. Third parameter is an integer value k. This function should *cyclically rotate* all the elements of the array arr, by k positions to the right.

For example,

if **arr** contains these 8 elements **{ 3, 4, 5, 8, 7, 2, 9, 1 }**,
then, after the function call **cyclicRotate (arr, 8, 3)**
the **arr** should contain these 8 elements **{ 2, 9, 1, 3, 4, 5, 8, 7 }**.

In your function:

- You can assume that k is greater than 0 and less than n
- You are NOT allowed to declare or use any other array.

Illustrate the working of your function (on at least 4 different examples/inputs) in a complete C program.
Hint 1: What happens if k is 1?
Hint 2: You may implement another helper function to accomplish this task.
Hint 3: Do some paperwork.

Question # 4

Modify question number 3 so that your program works fine if k > n (number of cyclic rotations is greater than size of array).

Question # 5

Write a C program to **delete elements from an array**.

```
Enter the size of the array: 5

Enter the 1 elements of the array: 1

Enter the 2 elements of the array: 2

Enter the 3 elements of the array: 3

Enter the 4 elements of the array: 4

Enter the 5 elements of the array: 5

Enter the element to be deleted: 3

Array after deleting 3 is: 1 2 4 5
```

PART B

Question # 6

Write C function your that display the $N \times N$ matrix (array) on screen in a neat and readable way.
After that, write another function that swaps the contents of the ***main diagonal*** (which runs from top-left to bottom-right corner) of the matrix with the contents of the ***antidiagonal*** (which runs from top-right to bottom-left corner). Finally, your program should once again display the (now modified) $N \times N$ matrix (array) on screen.

Important Note: You MUST implement the logic of your program using at least **2 different functions** (apart from the main function).
For example, if ***N*** is **7** then your program should display the following matrices on screen:

```
Initial matrix (BEFORE swapping diagonals):

  1   8  15  22  29  36  43
  2   9  16  23  30  37  44
  3  10  17  24  31  38  45
  4  11  18  25  32  39  46
  5  12  19  26  33  40  47
  6  13  20  27  34  41  48
  7  14  21  28  35  42  49


Final matrix (AFTER swapping diagonals):

 43   8  15  22  29  36   1
  2  37  16  23  30   9  44
  3  10  31  24  17  38  45
  4  11  18  25  32  39  46
  5  12  33  26  19  40  47
  6  41  20  27  34  13  48
 49  14  21  28  35  42   7
```

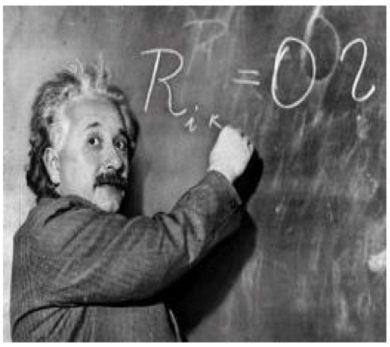
Note *that the elements on the* ***main diagonal*** *have been put in* BLUE *color, elements on the* ***antidiagonal*** *have been put in* GREEN *color, and the common element between the two diagonals (i.e. the center-most element) has been put in* RED *color. These colors have been used just for your understandability, and the output produced by your program will (obviously) not be colored.*

Question # 7

C program to print the upper triangular matrix.

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

Upper triangular matrix is:
9 8 7
  4 6
    3
```



I NEVER TEACH MY PUPILS; I ONLY ATTEMPT TO PROVIDE THE CONDITIONS IN WHICH THEY CAN LEARN.

[... ALBERT EINSTEIN ...]

... (14 MARCH 1879 – 18 APRIL 1955) ...