National University of Computer and Emerging Sciences

Assignment

For

Object-Oriented Programming

Course Instructor(s)	Dr. Khalid Hussain
Year	2023

FAST School of Computing

CLO-2

Pointers
- Introduction to pointers
- Declaring Pointers, The Address-of Operator (&)
- Pointers to Pointers, Debugging Pointers
- Dynamic memory allocation
- Passing and returning arrays from functions

Instructions:

1. Make a word document with the naming convention "SECTION_ LAB#_ROLLNO" and put all your source code and snapshots of its output in it. Make sure your word file is formatted properly.
2. Zip the word file with all Source code.
3. Plagiarism is strictly prohibited.
4. Do not discuss solutions with one another.

Question#1

Write a program that takes a pointer **ptr1** and dynamically allocate it with n size of array (Take n as an input from user), initialize it with value given by user. Take another pointer **ptr2** and allocate memory of n size (same as previous). Now make a function that takes both dynamic arrays as parameters and Store the reverse of **ptr1** into **ptr2.** Now check whether the **ptr2** is **Palindrome** or not. (Use Pointer arithmetic while reversing the dynamic arrays). In the end, also print both arrays using pointer notation.

Example Input/output:

Ptr1 input: 1,5,9,3,2

Ptr2: 2,3,9,5,1

HINT: If an array remains same when it is reversed, it is known as **Palindrome.**

Question#2

In linear algebra, a symmetric matrix is a square matrix that is equal to its transpose.

(a) Create a user-defined matrix using a 2D Dynamic Array.

Now, implement a function that takes this array as a parameter and finds out if it's a symmetric matrix or not.

Display both the originally entered 2D array, as well as the transpose of the 2D array. Also, display if it was symmetric or not. The memory leaks and dangling pointer must be avoided.

(b) Repeat the same task using single pointer instead of double.

Example Input/Output:

Given input:

1 2 3

4 5 6

7 8 9

Output:

Original Matrix:

1 2 3

4 5 6

7 8 9

Transpose:

1 4 7

2 5 8

3 6 9

Result: Not Symmetric.

Question#3

Write a user-defined function named upper_half() which takes a two-dimensional array as a parameter, and prints only the upper half of the array.

Example:

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

Question#4

Explain how Shallow copy and deep copy differs from each other. Write a piece of code in which you do shallow copy and then deep copy. After that explain the effects of both on our code. Is Deep copy better in some situations? If yes, explain why?

Question#5

Write a program that inputs the size of an integer array and then starts taking input of the elements of the array. If the number of elements exceed the size of the array, your program should extend the size of the array to accommodate those extra elements. The array name after extending the size will remains same. Make sure the memory leaks and dangling pointer must be avoided.

Implement functions to increase the size of your array.

Example Input/Output:

- Enter Array Size N
- Suppose user enters N = 5
- Enter Array Elements
- 1 2 3 4 5 6 7

- Array space for the first five elements was already allocated, a function should now be called to allocate extra space for any number of elements the user decides to enter after those 5. [i.e., the array size should now get modified to N=7]
- Display Array
- Display Size

NOTE: Your program must be generic and should work for any number of elements. There should be no hardcode that only works for a certain number/example.