University of Lincoln
School of Computer Science
CMP9133M – Advanced Programming
Workshop 4

Task1:
Try to answer most of the following brief questions in order to understand how pointers works under different situations (e.g., pointing at variables, passing them to functions, pointing at constants). You may create a C++ console application for each of the questions.

Question 1: Pointer Basics

1. Declare an integer variable `num` and initialize it with the value 42.
2. Declare a pointer named `ptr` that points to the address of `num`.
3. Using the pointer `ptr`, modify the value of `num` to 87.

Question 2: Dynamic Memory Allocation

1. Write a program to dynamically allocate memory for an integer.
2. Read an integer value from the user and store it in the allocated memory.
3. Display the value stored in the allocated memory.

Question 3: Array and Pointer Relationship

1. Declare an array of integers with a size of 5 and initialize it with values [10, 20, 30, 40, 50].
2. Declare a pointer named `arrPtr` and make it point to the first element of the array.
3. Using pointer arithmetic, modify the value of the second element to 25.

Question 4: Pointer to Pointer

1. Declare an integer variable `num1` and set its value to 100.
2. Declare a pointer `ptr1` that points to the address of `num1`.
3. Declare another pointer `ptr2` that points to the address of `ptr1`.
4. Using `ptr2`, modify the value of `num1` to 200.

Question 5: Passing Pointers to Functions

1. Write a function called `swapValues` that takes two integer pointers and swaps their values.
2. Test the function by passing two integer variables and printing their values before and after the function call.

Question 6: Pointers and Strings

1. Declare a character array containing the string "Hello, C++ Pointers!".
2. Declare a pointer to a character named `strPtr` and make it point to the start of the array.
3. Using pointer arithmetic, print each character of the string one by one until the null terminator is reached.

Question 7: Pointer to Constant

1. Declare a constant integer variable `constVar` with the value 50.

2. Declare a pointer named `constPtr` that points to `constVar`.
3. Try to modify the value of `constVar` through `constPtr` and explain the outcome.

Question 8: Pointer Arithmetic

1. Declare an integer array with values [10, 20, 30, 40, 50].
2. Declare an integer pointer `intPtr` that points to the third element of the array.
3. Perform pointer arithmetic to access the second and fourth elements using `intPtr`.

Task2:

You are required to implement a program that manages a dynamic array of integers. Your program should allow users to perform various operations on the array, including inserting elements, deleting elements, and displaying the contents of the array.

Instructions:

1. Define a class named `DynamicArray` with the following private attributes:
    - An integer pointer named `array` to store the dynamic array.
    - An integer variable named `size` to store the current size of the array.
2. Implement the following public methods for the `DynamicArray` class:
    - A constructor that initializes the `array` pointer to `nullptr` and the `size` to 0.
    - A destructor that deallocates memory for the dynamic array.
    - `void insert(int value)`: Adds the given value to the end of the array.
    - `bool remove(int value)`: Removes the first occurrence of the given value from the array.
    - `void display()`: Displays the elements of the array.
3. In the `main()` function:
    - Create an instance of the `DynamicArray` class.
    - Use a loop to repeatedly display a menu of options and allow users to choose operations on the dynamic array. The menu options should include:
        - Insert an element.
        - Remove an element.
        - Display array.
        - Exit.

Note: Make sure to handle memory allocation, deallocation, and dynamic array resizing correctly in your `DynamicArray` class.

You can use the code provided on blackboard called `dynamic_array_manipulation.cpp`. This code provides the basic structure for the "Dynamic Array Manipulation" problem and includes the `DynamicArray` class and the main menu loop. Fill in the missing parts to complete the implementation and achieve the desired functionality.