Lab02

In a main.cpp file, implement the following functions:

Task 1

- Write a function PointerExample() that:
    1. Declares an integer variable and assigns it a value.
    2. Declares a pointer to an integer.
    3. Stores the address of the variable in the pointer.
    4. Prints:
        - The value of the variable
        - The address of the variable
        - The value stored in the pointer (the address)
        - The value pointed to by the pointer

This will help you understand how variables, addresses, and pointers are related.

Task 2: Printing an Array Using Pointers

- Write a function printArray(int* arr, int size) that:
    1. Takes a pointer to an array and its size as parameters.
    2. Prints the array in the format [1,2,3,...].
    3. Uses **pointer arithmetic** (e.g., *(arr + i)) to access elements, not arr[i].

Task 3: Dynamic Array from User Input

- Write a function getScores(int& size) that:
    1. Prompts the user to enter the number of scores.
    2. Validates that the size is positive.
    3. Dynamically allocates an array of integers using new.
    4. Prompts the user to enter each score and stores them in the array.
    5. Returns the pointer to the array.

Remember to return the **pointer** to the newly created array.

Task 4: Joining Two Arrays

- Write a function Join(int arr1[], int size1, int arr2[], int size2) that:
    1. Dynamically allocates a new array large enough to hold both arrays.
    2. Copies the elements of arr1 into the new array.
    3. Copies the elements of arr2 after the elements of arr1.
    4. Returns a pointer to the new combined array.

Task 5: Main Program

- In your main() function:
    1. Call PointerExample() (Task 1).
    2. Use getScores() to create a dynamic array of scores.
    3. Print the scores using printArray().
    4. Create two sample arrays (e.g., {1,2,3} and {4,5,6,7}) and call Join() to combine them.
    5. Print the combined array using printArray().
    6. Use delete[] to free all dynamically allocated memory.