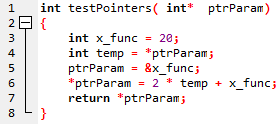
**Problem 20.1**

Consider the following function receiving a (non-constant) pointer to integer parameter.

****

1. Call this function in your program and print the value returned by this function. Observe the output and discuss if there is any confusion.
2. Change the line 1 of above function to following and compile your program again to check for output/error.
3. ****
4. ****
5. ****

**Problem 20.2**

Write down a function called getPolarity(), which receives a 1D array and its size. The function converts all positive values in the array to +1 and all negative values in the function to -1. However, this change should not be made into parameter array and it should remain unchanged. Rather, a new array should be created and returned from the function containing the above changes to original array. Note that the parameter array elements with 0 value will remain unchanged, i.e. they will still have the 0 value in the output array.

**Problems from previous lab**

**Problem 19.2**

Write down a function called doubleIt, whose function is to simply multiply every value of the array passed to it with 2. Call this function in main and print the values of the array before and after calling the function to verify that arrays are always passed by reference.

**Problem 19.3**

Hint: Accessing array elements using pointer/offset notation: point the pointer to starting address of array, then use \*(ptr +i) to access the value of ith element of the array.

Complete the following tasks using pointer notation to access array.

1. Input an array using pointer/offset notation and then print the array using pointer/offset notation.
2. Input an array using pointer/offset notation and print the array in reverse using pointer/offset notation.

**Home Task**

1. Implement the quadratic roots program using call by reference, i.e. Write down a function called computeRoots(int a, int b, int c, int \*x1, int \*x2), which should compute the two roots using quadratic equation and return their values in x1 and x2, which are passed by reference.
2. Implement the array linear search algorithm in a function called linearSearchUsingPointer using pointer/offset notation, i.e. the function should print whether a key value is found inside the array or not, but this should be done using pointer/offset notation.