**Problem 1:**

Create a program that counts the occurrence of the word ‘that’ in a file (use the string class’ operator ‘==’ to find the word)

**Problem 2:**

Create a vector <float> and put 25 numbers into it. Then square each number and put the result back into the same location in the vector. Display the vector before and after the multiplications.

**Problem 3:**

Create two functions, one that takes a string* and one that takes a string&. Each of these functions should modify the outside string object in its own unique way. In main(), create and initialize a string object, print it, then pass it to each of the two functions, printing the results.

**Problem 4:**

Create a struct that holds two string objects and one int. Use a typedef for the struct name. Create an instance of the struct, initialize all three values in your instance, and print them out. Take the address of your instance and assign it to a pointer to your struct type. Change the three values in your instance and print them out, all using the pointer.

**Problem 5:**

Given an array prices, where the i-th element prices[i] represents the price of a given stock on day i.

You are only allowed to choose **one day** to buy this stock and choose **a different day in the future** to sell it.

Design an algorithm to calculate the maximum profit you can achieve.

Return the maximum profit you can obtain from this transaction. If no profit can be achieved, return 0.

**Example 1:**

Input: [7,1,5,3,6,4]

Output: 5

Explanation: Buy on day 2 (price = 1) and sell on day 5 (price = 6), profit = 6 - 1 = 5.

Note that profit cannot be 7 - 1 = 6, because the selling price must be greater than the buying price, and you cannot sell before buying.

**Example 2:**

Input: prices = [7,6,4,3,1]

Output: 0

Explanation: In this case, no transaction is completed, so the maximum profit is 0.

**Constraints:**

1 <= prices.length <= 10^5

0 <= prices[i] <= 10^4