1. Write a function match which takes 2 integer arrays (named one and two) and returns the number of times "matches" occur in parallel positions in the two arrays. That is, count the number of times

one[i] == two[i]. The size of both arrays is the same. The arrays

are passed as parameters along with their size.

**int match(int one[], int two[], int size1, int size2)**

**{**

**int count = 0;**

**int min\_size;**

**if (size1 < size2) {**

**min\_size = size1;**

**} else {**

**min\_size = size2;**

**}**

**for (int i = 0; i < min\_size; ++i) {**

**count += (one[i] == two[i]);**

**}**

**return count;**

**}**

1. Write a prototype for your function in question 1.

**int match(int one[], int two[], int size1, int size2);**

1. Write the statements to declare two arrays of size 25, read values into the arrays, call your match function, and print out the number of matches in the two arrays.

**int array1[25] = {};**

**int array2[25] = {};**

**for (int i = 0; i < 25; ++i) {**

**cin >> array1[i];**

**}**

**for (int i = 0; i < 25; ++i) {**

**cin >> array2[i];**

**}**

**cout << "Number of matches: "**

**<< parallel\_matches(array1, array2, 25, 25) << endl;**

1. Write a function named count that will count and return the occurrences of a given character in an array named letters. The parameters will be the array letters, the size of the array, and the character to count.

For example: If the array contained the values

x 8 R A a a 0 s S a A

and the character to count was 'a', then the function would return the value 3.

**int count(char characters[], int size, char count\_char)**

**{**

**int count = 0;**

**for (int i = 0; i < size; ++i) {**

**count += characters[i] == count\_char;**

**}**

**return count;**

**}**

1. Write a prototype for the function in the previous problem.

**int count(char characters[], int size, char count\_char);**

1. Write the statements to declare an array of characters, and initialize the array to contain the characters: f A i @ N Z a 7 p Y h A. Call the count function and print the number of times the character p is contained in the array.

**char characters[] = {'f', 'A', 'i', '@', 'N', 'Z',**

**'a', '7', 'p', 'Y', 'h', 'A'};**

**cout << "'p' appears " << count(characters, 12, 'p') << " times"**

**<< endl;**

1. Write a function called search that receives an array of ints, the size of the array, and a number to search for. The function will return true if the number is contained in the array, and will return false otherwise.

**bool search(int nums[], int size, int element)**

**{**

**for (int i = 0; i < size; ++i) {**

**if (nums[i] == element) {**

**return true;**

**}**

**}**

**return false;**

**}**

8. Write a prototype for your search function.

**bool search(int nums[], int size, int element);**

9. Write the statements to call your function to search for the value

1500 in an array called salaries which has 100 elements. Print a

message telling whether the value was found in the array.

**if (search(salaries, 100, 1500)) {**

**cout << "1500 was found";**

**} else {**

**cout << "1500 was not found";**

**}**

**cout << endl;**