

## CSCI 201 – Computer Science 1

### Lab assignment 8: Searching for a matching-sum in an array.

To be completed by Wednesday March 20.

**Objectives:** Apply the concept of employing a nested `for` loop for processing an array. Design a nested `for` loop for the Matching-sum problem.

The **Matching sum** problem is defined as follows: *Given an array,  $A$ , containing  $n$  items in non-decreasing order, and an item,  $x$ , find out if there are 2 distinct indices  $i$  and  $j$ , such that  $A[i] + A[j] = x$ .  $A[i]$  and  $A[j]$  together constitute a **Matching Sum** for  $x$ . (This problem has widespread applications in several areas of computing.)*

We need a program that does the following:

- (1) Reads a file containing several numbers in non-decreasing order and stores these in the array.
- (2) Prompts the user to input a number for which a matching sum is needed.
- (3) Checks for a matching sum. If a matching sum is found, then the indices  $i$  and  $j$  and the values in  $A[i]$  and  $A[j]$  are reported; if no matching sum is found, the program reports failure. The program also prints a count of the total number of “array access operations” (every time a pair of square brackets appears in the program, count it as an array access operation).

**Sample run:** The array contains the numbers: 2 3 3 4 7 9 13 16 16 Two sample runs are given below.

```
Please enter the number for which a matching sum is needed: 23
Item 4 (with value 7) and and item 7 (with value 16) add up to 23.
58 array items were accessed
```

```
Please enter the number for which a matching sum is needed: 35
No matching sum was found
72 array items were accessed
```

*For the lab, we use the “brute force” strategy: make several passes over the array, so that we check every pair  $(A[i], A[j])$ , and compare the sum  $A[i]+A[j]$  against  $x$ . The loops should be exited as soon as the matching sum is found. (See the handout on nested loops.)*

**Question 1:** Think about how we should make passes over the array. List the order in which all the pairs of items will be examined in each pass, assuming an array of size  $n$ . (see the handout on **Designing Nested for Loops** to see how this is done.)

**Question 2:** What action should we take for each pair of items that we are examine?

**Question 3:** Write a sequence of simple `for` loops for each pass. (see the handout on **Designing Nested for Loops** to see how this is done.)

**Question 4:** Roll all the loops from your answer to Question 2 into a single nested `for` loop. (see the handout on **Designing Nested for Loops** to see how this is done.)

**Question 5:** Implement a C++ program that implements your nested loop and terminates the loops with a `break` statement when a matching sum is found. Use a global variable to keep track of the number of array accesses. Run your program using a file with 20 items. Test multiple values of  $x$ , some with a matching sum and some without. Create a script file.

**Question 6:** In the test runs, how many array accesses were needed to verify that no matching sum exists?

Upload the script for Question 5, and answers to the other questions.