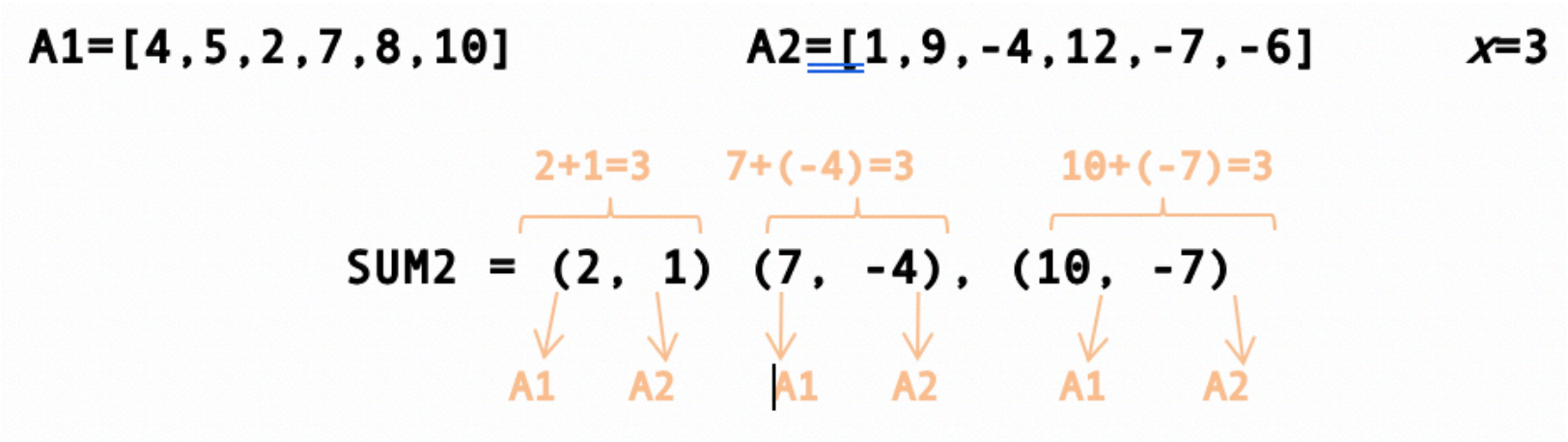
Lab Assignment

(based on exercise 2.3.7 in your textbook)

- SUM2 Problem Definition

  Given two unsorted arrays **A1** and **A2**, and an integer ***x***, find the set **SUM2** consisting of all pairs of values (**a1**, **a2**), where **a1** ∈ **A1** and **a2** ∈ **A2**, such that **a1+a2=*x***.

- Example:



Algorithm

```
SUM2ALGORITHM (A1, A2, x)
1. SUM2 = {}
2. For each value, v1, in A1:
2.    v2 = x - v1
3.    Search array A2 for value v2. If v2 is found in A2:
4.          Add the pair (v1, v2) to SUM2
5. return SUM2
```

Assignments
(to be completed in teams)

```
SUM2ALGORITHM (A1, A2, x)
1. SUM2 = {}
2. For each value, v1, in A1:
2.    v2 = x - v1
3.    Search array A2 for value v2. If v2 is found in A2:
4.          Add the pair (v1, v2) to SUM2
5. return SUM2
```

From the attached document:

1. Complete TASK 1 (running time analysis) of the attached document as lab assignment. You have 45 minutes to do it and then you will present it to the class.
2. Complete TASK2 (implementation) and TASK 3 (testing) as homework assignment.

For the **Search** in line 3, you will use the following search algorithm:

- Students in groups 1, 4 – Linear/Sequential Search
- Students in groups 2, 5 – Binary Search
- Students in groups 3, 6 – Hash-based Search

Assert Example

You have to include the *cassert* library at the top of your program:

- The assert function is useful for detecting violations of a precondition.
- If the test succeeds (TRUE), the program continues
- If the test fails (FALSE), the program stops

```
#include <cassert>
double sqroot( double x)
  //Returns the square root of x
  //Precondition:  x  >=  0
  {
       assert(x >= 0);

     • • •

  }
```