NAME: _____

Learning Objectives

- Learn how to use vectors and arrays.
- Learn how to write algorithms that perform sequential (linear) processing.

Instructions

Work out the answers to these problems manually without the use of a computer. This will help you develop the ability to read and analyze code.

Another reason to practice solving these problems manually is that problems of this type will appear on Quiz 3 and subsequent quizzes and exams. During the quiz/exam, you will not have access to a computer to solve these problems, so you need to develop the ability to read code, analyze it and think critically.

After coming up with answers manually, you can write and run test code to check whether you have solved the problems correctly.

Submit a single document with your answers either by email or through your remote git repository. If submitting through git, your document should be located in a folder named *a3*.

Problems

```
bool sameLength(const vector<int> & a, const vector<int> & b);
```

Figure 1

1) Write a predicate function called *sameLength* that checks whether two vectors have the same number of elements. A declaration of the function is shown in Figure 1. The function takes references to 2 vectors as arguments and returns true if the two vectors have the same size and false otherwise. (10 points)

```
bool isIdentical(const vector<int> & a, const vector<int> & b);
```

Figure 2

2) Write a predicate function that checks whether two vectors are identical (contain exactly the same elements in the same order). A declaration of the function is shown in Figure 2. The function returns true if the two vectors are identical, otherwise it returns false. (10 points)

```
bool isNonDecreasing(const vector<int> & v);
```

Figure 3

3) Write a predicate function called *isNonDecreasing* that checks whether a vector of integers contains values that are in non-decreasing order. A declaration of the function is shown in Figure 3. The function returns true if the elements are in non-decreasing order, otherwise it returns false. For example, it will return true for v = (3, 4, 4, 6, 8), and it will return false for (3, 4, 6, 2, 5). (10 points)

```
void addThree(int a[ROWS][COLS]);
```

Figure 4

4) Suppose that array of int *a* has *ROWS* rows and *COLS* columns, where *ROWS* and *COLS* are some constants. Implement a function called *addThree* that uses a loop nested inside another loop to add the value 3 to each element of the array. A declaration of the function is shown in Figure 4. (10 points)

The following is an example declaration of *ROWS* and *COLS*.

```
const int ROWS = 3;
const int COLS = 2;
```

```
void randomFill(vector<int> & v);
```

Figure 5

5) Implement the function *randomFill* declared in Figure 5 so that it fills the vector *v* passed into it with 100 distinct random numbers. Distinct means that a given number can appear at most once in *v*. Use the function *rand()* to get random integers. Assume that *v* is empty when the function is called. (10 points)