CS 180
Test 3 – 6 November 2019

Name _____

1. (12 pts) Write the definition of a function that finds and returns the *position* of the minimum element in an unsorted vector of int values. Do not write a prototype or any Javadoc, just the function itself.

2. (10 pts) Given the following declaration, state what each output will be:
```
unsigned numbers[] {2, 4, 6, 8, 10};
```

   (a)
```
cout << numbers[2];
```

   (b)
```
cout << numbers[0] + 2;
```

   (c)
```
int x = numbers[1];
x--;
numbers[4] = x;
cout << numbers[2] << ' ' << numbers[3] << ' ' << numbers[4];
```

3. (10 pts) Assume the following declarations have already been made:
```
int array1[SIZE];
int array2[SIZE];
```
and assume some code has already been run that fills `array1` with data. Write a code fragment that will copy the entire contents of `array1` into `array2`.

4. (15 pts) Given the following array

3	4	7	11	14	16	23

for each of the following values, in performing a binary search, for each value below, show the exact sequence of values taken on by the variable `middle` in searching for the value.

(a) 2

(b) 18

(c) 7

5. (xx pts) Write the Javadoc, prototype, and definition of a function that accepts a measured length in inches and returns the equivalent length in centimeters. One inch equals 2.54 centimeters.

6. (12 pts) Write the Javadoc and the prototype of a function named `display_values` that will be used to print all the values of an array of doubles that are currently in use. Do *not* write the function itself, only the prototype. Then, write a single statement that would appear in `main` that calls your function.

7. (10 pts) Given the following code, draw a picture of memory when line 15 has just finished running, and then show what would change by the time line 17 has just finished running by lightly crossing out the previous value(s) and showing the new values. Be sure to diagram the memory for both `main` and for `double_value`. Finally, show the output when the code is run to completion.

```
1  int main()
2  {
3    int array[] {3, 8, 9, 5, 4};
4
5    increment_a_value(array, 3);
6    for (auto a : array)
7    {
8      cout << a << ' ';
9    }
10   cout << endl;
11   return 0;
12 }
13
14 void increment_a_value(int values[], size_t position)
15 {
16   values[position]++;
17 }
```