1 Arrays

1. What is the value of `result` when the following code is executed?

```
    int[] taxicab = {1, 7, 2, 9};
    int result = taxicab.length;
```

2. What is the value of `result` when the following code is executed?

```
    int[] taxicab = {1, 7, 2, 9};
    int result = taxicab[1] + taxicab[3];
```

3. What is the value of `result` when the following code is executed?

```
    int[] taxicab = {1, 7, 2, 9};
    int result = taxicab[taxicab.length - 1];
```

4. What is the value of `result` when the following code is executed?

```
    int[] taxicab = {1, 7, 2, 9};
    int result = 0;
    for(int i = 0; i < taxicab.length; i++) {
        result+=taxicab[i];
    }
```

5. What is the state of array `taxicab` when the following code is executed?

```
    int[] taxicab = {1, 7, 2, 9};
    int result = 0;
    for(int i = 0; i < taxicab.length; i++) {
        taxicab[i]*=2;
    }
```

6. What is the state of array `taxicab` when the following code is executed?

```
int[] taxicab = {1, 7, 2, 9};
int result = 0;
for(int i = 0; i < taxicab.length; i++) {
    if(i % 2 == 0) {
        taxicab[i]*=2;
    }
}
```

$\{2, 7, 4, 9\}$

7. What is the state of array `taxicab` when the following code is executed?

```java
int[] taxicab = {1, 8, 6, 10, 9, 5, 7};
int result = 0;
for(int i = 0; i < taxicab.length; i++) {
        if(taxicab[i] % 2 == 0) {
                taxicab[i]/=2;
        }
}
```

8. Write a piece of code that declares and instantiates a array that can hold 8000 floating-point values.

9. Write a piece of code that declares and instantiates an array `arr` that can hold 666 boolean values.

10. Assuming that the array `arr` holds an array that holds 2000 integers (that is, it has already been declared and instantiated), write a piece of code, that, using a loop, assigns,

    - 1 to the first item of the array
    - 2 to the second item of the array
    - 3 to the third item of the array
    - ...

11. Assuming that the array `arr` holds an array that holds 2000 integers (that is, it has already been declared and instantiated), write a piece of code, that, using a loop, assigns,

    - 1 to the first item of the array
    - 5 to the second item of the array
    - 9 to the third item of the array
    - 13 to the fourth item of the array
    - ...

12. Assuming that the array `arr` holds an array that holds $n > 0$ integers (that is, it has already been declared and instantiated), write a piece of code, that, using a loop, assigns,

    - n to the first item of the array
    - $n - 1$ to the second item of the array
    - $n - 2$ to the third item of the array
    - ...
    - 1 to the last item of the array

    Note that you can access the number of items in array `arr` by `arr.length`.

13. Consider the following array `arr`,

```java
        float[] arr=  {-1.2, 2.5, 1.3, 0, 0, 1.7, -1.9, 1.1, 0, 0, 0.6};
```

(a) Write a piece of code that stores in a variable `result`, the number of items in array `arr` that are greater than 1.4.

(b) Write a piece of code that stores in a variable `result`, the number of negative items in array `arr`.

(c) Write a piece of code that stores in a variable `max`, the highest value stored in array `arr`.

14. Assuming that array `arr` hold 20 random integers, write a piece of code that stores in a variable `result`,

    - `true` if the array `arr` is sorted in ascending order (such that each item is more than or equal to the previous item).
    - `false` otherwise.

15. **(challenging)** Assuming that array `arr` hold 20 random integers, write a piece of code that stores in a variable `allUnique`,

    - `true` if every item in the array `arr` is unique.
    - `false` otherwise.