1 Functions - 2

1. Consider the following definition for function `foo`.

```
int foo(int[] a) {
        int c = 0;
        for(int i = 0; i < a.length; i++) {
                if(a[i] == 0) {
                        c++;
                }
        }
        return c;
}
```

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

```
        int[] arr = {6,0,9,0,0,8,0,8,9,7,0,0};
        int result = foo(arr);
```

2. Consider the following definition for function `foo`.

```
boolean foo(int[] a, int target) {
        for(int i = 0; i < a.length; i++) {
                if(a[i] == target) {
                        return true;
                }
        }
        return false;
}
```

   (a) What is the value of `result` when the following code is executed?

```
        int[] arr = {2,1,3,6,4,9,7,8};
        boolean result = foo(arr, 5);
```

   (b) What is the value of `result` when the following code is executed?

```
        int[] arr = {2,1,3,6,4,9,7,8};
        boolean result = foo(arr, 9);
```

3. Consider the following definition for function `foo`.

```
boolean foo(int[] a) {
        for(int i = 0; i < a.length; i++) {
                if(a[i]%2 == 1) {
                        return false;
                }
        }
        return true;
```

```
}
```

Write a piece of code that calls the function `foo` by passing an integer array `arr` and storing the value returned in a variable `result`. You may assume the array `arr` has at least one item in it.

4. Consider the following definition for function `foo`.

```
boolean foo(int[] a, int [] b) {
        if(a.length != b.length) {
                return false;
        }
        for(int i = 0; i < a.length; i++) {
                if(a[i] != b[i]) {
                        return false;
                }
        }
        return true;
}
```

   (a) Write a piece of code that calls the function `foo` by passing two integer arrays `myArray, yourArray` and storing the value returned in a variable `result`. You may assume the arrays `myArray, yourArray` both have at least one item each.

   (b) Explain briefly what is the purpose of function `foo`.

5. Write a function that when passed an array containing floating-point values, doubles each item of the array. Remember that modifying the formal array inside a function modifies the actual array passed.

6. Write a function that when passed an array containing floating-point values, negates each negative item of the array so that it becomes positive.

7. Write a function that when passed an integer array, divides every even number in the array by 2.

8. **(somewhat challenging for most)** Write a function that when passed an integer array, resets any composite (non-prime) number to 0. You may define a *helper* method.

9. **(somewhat challenging for most)** Write a function that when passed an integer array, returns an array with only the positive items from that array.