String Lab

1) Consider the following method. What value is returned by the call `eval(“jamjarjax”, “ja”)?`

```
int eval(String str, String check) {
  int m = str.length()/2;
  String a = str.substring(0, m);
  String b = str.substring(m);
  return a.indexOf(check) + b.indexOf(check);
}
```

2

2) Consider the following code. What will be printed?

```
void mystery(String str) {
  if (str.length() < 4) {
    println(“D”);
  }
  else {
    print(str.substring(0, 1));
    mystery(str.substring(1));
    print(str.substring(0, 1));
}

mystery(“BELLE”);
```

BEDEB

3) Write a function `int lastIndexOf(String str, String substr)`, which returns the starting index of the last occurrence of `substr` in `str`.

```
int lastIndexOf(String str, String substr) {
  boolean ended = false;
  int index = -1;

  while (!ended) {
    int tempIndex = str.indexOf(substr,index+1);
    if(tempIndex == -1) {
       ended = true;
    } else {
       index = tempIndex;
    }
  }
  return index;
}
```

4) Write a program that splits the numbers in the given `myNums` string, converts them to floats, and prints them to the console. You may assume that all the numbers are comma+single-space separated and they are all floats. However, your code should work for arbitrary many numbers and numbers with an arbitrary number of integer and floating-point positions.

```
void setup() {
String myNums = "1.2345, 2.3, .345, 4.0, 5.123345678";

  // Add your code here
  String[] nums = myNums.split(", ");
  for(int i = 0; i < myNums.length; ++i){

        println(float(nums[i]));

  }

}
```

5) Finish the following program, which was designed to count and print the number of duplicate strings in the `myArray` string array.

```
// Count and print the number of duplicates in myArray
String[] myArray = {"A", "B", "C", "D", "A", "F", "C"};
void setup() {
  int count = 0;

  // Add code here

  for (int i = 0; i < myArray.length; ++i) {
  for (int j = i+1; j < myArray.length; ++j) {
    if (myArray[i] != null &&
      myArray[i].equals(myArray[j])) {
      ++count;
      myArray[j] = null;
    }
  }
}

  println("There are " + count + " duplicates.");
}
```

6) Write a recursive function `boolean palindrome(String str)` that takes a `String` argument `str` and returns `true` if `str` is a palindrome and `false` otherwise.

```
boolean palindrome(String p) {
  if (p.length() <= 1) {
    return true;
  } else if (p.charAt(0) != p.charAt(p.length()-1)) {
    return false;
  } else {
    return palindrome(p.substring(1, p.length()-1));
  }
}
```