CS 106 Winter 2019

Lab 02: Arrays and Strings

Summary

This lab will allow you to practice with manipulation of arrays and strings, and the relationship between the two. Each question is on a separate page.

SAVE each sketch as "L02Q01", "L02Q02, "L02Q03", etc.

QUESTION ONE

Given the following starter code, write the function `repeat()`.
- `repeat()` has two parameters.
  - The first parameter is of type `String`.
  - The second parameter is of type `int`. It is always at least 1.
- `repeat()` returns a string.

`repeat()` that takes the two values as input and returns a new `String` in which the provided string has been repeated the given number of `times`.

For example, `repeat("cat", 5)` would return `"catcatcatcatcat"`.

```
String orig = "Help";
int reps = 3;
String result;

void setup() {
  result = repeat(orig, reps);
  println("The original string was: " + orig);
  println("The new string is: ", result);
}
```

After you create the function `repeat()`, the code above should produce the result below.

The original string was: Help

The new string is: HelpHelpHelp

QUESTION TWO

Write a function `isIncreasing()` that takes an array of integers as input and returns a boolean value. The function checks if the array of integers is *strictly increasing*: that is, if each number in the array is strictly larger than the one before it. For example, the array { -3, 4, 9 } is strictly increasing, but { 1, 4, 2 } and { 3, 5, 5, 6 } are not.

Use the following starter code:

```
int[] nums = {1, 2, 5, 8, 7};
boolean increase;

void setup() {
  increase = isIncreasing(nums);
  if (increase) {
    println("The array is strictly increasing");
  } else {
    println("The array is not strictly increasing");
  }
}
```

After you create the function `isIncreasing()`, the code above should produce the result below.

```
The array is not strictly increasing
```

QUESTION THREE

Write a function `countOccurrences()` that takes two values as input: an array of `String`s and a single search `String` in a separate parameter. The function returns the number of times that the search string occurs in the array. For example, if `arr` holds the array `{ "apple", "pear", "peach", "apple", "apple" }`, then `countOccurrences( arr, "apple" )` would return 3, and `countOccurrences( arr, "durian" )` would return 0.

```
String[] fruits = { "apple", "pear", "peach", "apple", "apple" };
String s = "apple";
int numFound;

void setup() {
  numFound = countOccurrences(fruits, s);
  println("The number of " + s + " found is " + numFound);
}
```

After you create the function `countOccurrences()`, the code above should produce the result below.

```
The number of apple found is 3
```

QUESTION FOUR

Write a function `doubleOrNothing()`. It takes an array of integers as input and returns a new array of integers, of the same length as the input. In the output, every positive number in the input array is doubled, and every negative number in the input array is replaced by zero. For example, if `arr` holds the array { `1, 2, -5, 8, -3, 12` }, then `doubleOrNothing(arr)` would return the array { `2, 4, 0, 16, 0, 24` }.

Use the following starter code.

```
int[] arr1 = { 1, 2, -5, 8, -3, 12 };
int[] arr2;

void setup() {
  arr2 = doubleOrNothing(arr1);
  println("The resulting array is:");
  printArray(arr2);
}
```

After you create the function `doubleOrNothing()`, the code above should produce the result below.

```
0 1

1 2

2 -5

3 8

4 -3

5 12

The resulting array is:

[0] 2

[1] 4

[2] 0

[3] 16

[4] 0

[5] 24
```

QUESTION FIVE

Write a function `explode()` that takes a `String` as input and returns an array of characters (i.e., `char[]`) consisting of the characters in the order they originally appear in the string. For example, `explode("Dirk")` should return the array `{'D', 'i', 'r', 'k'}`.

Use the following starter code:

```
String s = "Hello World";
char[] c;

void setup() {
  c = explode(s);
  println("The character array is:");
  printArray(c);
}
```

After you create the function `explode()`, the code above should produce the result below.

```
The character array is:

[0] 'H'

[1] 'e'

[2] 'l'

[3] 'l'

[4] 'o'

[5] ' '

[6] 'W'

[7] 'o'

[8] 'r'

[9] 'l'

[10] 'd'
```

QUESTION SIX

Write a function `assemble()` that behaves like `explode()` in reverse. It takes an array of characters as input and returns a `String` consisting of all the characters in the array glued together. For example, if we have: `char[] cs = { 'D', 'i', 'r', 'k' };` then assemble() should return the `String` "Dirk".

Use the following starter code.

```
char[] cs = {'D', 'i', 'r', 'k'};
String s ;

void setup() {
  s = assemble(cs);
  println("The string is: " + s);
}
```

After you create the function `assemble()`, the code above should produce the result below.

```
The string is: Dirk
```

QUESTION SEVEN

The following code shows a string of text on the screen bounded by four lines. The resulting
screen is shown. Modify the following line of code in `draw()`
```
text( myText, 50, 100 );
```
to the following code:
```
text( myText, mouseX, mouseY );
```

Now, modify the four calls to line() so that the output displays at `mouseX, MouseY` and mimics
what you see in this youtube video: https://youtu.be/zBr3WEqKvHo

CS106 Lab 02
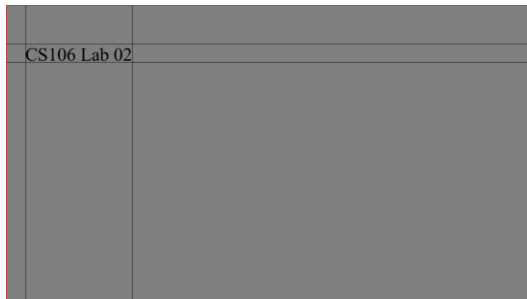
```
PFont fnt1;
String myText = "To be or not to be.";
int myFontSize = 48;

void setup(){
  size( 600, 600 );
  fnt1 = createFont("Times New Roman", myFontSize );
  fill( 0 );  // black
  textAlign( LEFT, TOP );
  textFont( fnt1 );
}

void draw() {
  background(127);
  text( myText, 50, 100 );
  // Top horizontal line.
  line(0, 100, width, 100);
  // Bottom horizontal line.
  line(0, 100+myFontSize, width, 100+myFontSize);
  // Left vertical line.
  line(50, 0, 50, height);
  // Right vertical line.
  line(50+textWidth(myText), 0, 50+textWidth(myText), height);
}
```

QUESTION EIGHT

This is meant to be an easy question.

The following code produces the following result. You have two tasks:
- Document the code.
- Change the three fonts to three different fonts of your choice. That is, change "Segoe Print", "Stencil", and "Yu Gothic UI Semilight" to three different fonts.

```
PFont font;

void setup() {
  size(400, 400);
  fill(0); // black
  printArray(PFont.list());
  text("The number of fonts is: " + PFont.list().length, 50, 50);
  text("The first Font is: " + PFont.list()[0], 50, 75);
  text("The last Font is: " + PFont.list()[PFont.list().length-1],
    50, 100);

  font = createFont("Segoe Print", 32);
  textFont(font);
  text("word1", 50, 150);

  font = createFont("Stencil", 32);
  textFont(font);
  text("word2", 50, 250);

  font = createFont("Yu Gothic UI Semilight", 32);
  textFont(font);
  text("word3", 50, 350);
}
```
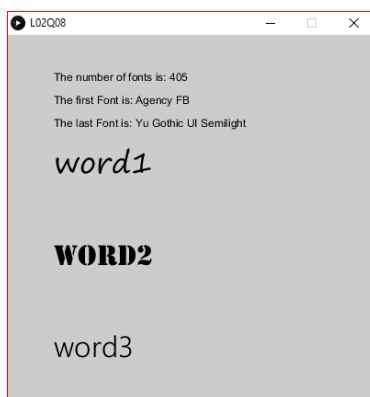
QUESTION NINE

This question relates to Assignment 02.

Given the following starter code, write the `draw()` function to produce the result shown below. Each element in the char[] array "pairs" is drawn evenly spaced across the screen. The black rectangular base is below each `char[]`. If the corresponding Boolean array element in the array "base" has the value true then the base is displayed and if the corresponding Boolean element is false the base is not displayed.

```
char[] pairs = {'a', 'c', 'b', 'd', 'd', 'a', 'b', 'c'};

boolean[] base = {true, true, true, false, true, false, true, true};

void setup() {

  size(800, 300);

  textSize(30);

  textAlign(CENTER, TOP);

}
```

Submission

Submit all sketch directories from this lab as one ZIP file called L02.zip to the lab dropbox on Learn.

It is your responsibility to submit to the correct dropbox with the correct files before the deadline. Otherwise you will receive a mark of 0.