KDDA: Web Front End Test

January 19, 2017

Submission instructions:

- *Answers to questions from Q1 to Q5 should be contained in a file named q1-q5.txt.*
- *Refer to DELIVERABLE section indicated on each coding exercise (Q6-Q10) for information on required file submissions. Your code must compile and run correctly.*
- *Compress your answer files in a single .zip container and send it as email attachment on or before the given deadline.*

Q1.

Consider the following Javascript function:

```
function foo ()
{
      return 5;
}
```

What will the following code do?

```
var myVar = foo;
```

PICK ONE OF THE CHOICES

A. Assign the integer 5 to the variable myVar
B. Assign the pointer to function foo to myVar
C. Throw an exception
D. Do nothing

Q2.

What is the role of the third parameter (async) of the XMLHttpRequest open method?

PICK ONE OF THE CHOICES

A. If true, the request is handled synchronously.
B. If true, the javascript engine is blocked while making the request
C. If true, the request can be handled without waiting.
D. If true, the request object is destroyed when 'send' is executed

Q3.

Consider the following CSS3 code:

```
div {
    border: 2px solid;
    resize: horizontal;
    overflow: auto;
}
```

What is the job of resize attribute here?

PICK ONE OF THE CHOICES

A. To make the div element resizable by the user.
B. To make the div element resizable by the browser.
C. To increase the horizontal padding.
D. To make the div element resizable by javascript.

Q4.

What is the correct way to add an element value to the end of a Javascript array?

PICK ONE OF THE CHOICES

A. arr[arr.length+1] = value
B. arr[arr.length] = value
C. arr[arr.length-1] = value
D. arr = arr + value

Q5.

Consider the two functions below. Will they return the same thing? Select the most accurate response.

```
function foo1()
{
  return {
      bar: "hello"
  };
}

function foo2()

{

  return

  {

      bar: "hello"

  };

}
```

PICK ONE OF THE CHOICES

A. yes
B. no
C. foo2 will give a run time error
D. foo1 will give a run time error

Q6.

Create a *mergeStrings()* function with *2* input parameters:

1. A string, *a*.
2. A string, *b*.

Your function must merge strings *a* and *b*, and then return a single merged string.

Constraints

- Max length of string parameters, a and b, is 25000.

Sample Input 0

```
abc
def
```

Sample Output 0

```
adbecf
```

Sample Input 1

```
ab
zsd
```

Sample Output 1

```
azbsd
```

DELIVERABLE

A Javascript file, q6.js, containing the *mergeStrings()* function defined by the stub below:

```
/*
 * Complete the function below.
 */
function mergeStrings(a, b) {

}
```

Q7.

Braces in a string are considered to be *balanced* if the following criteria are met:

- For every opening brace (i.e.: `(`, `{`, or `[`), there is a matching closing brace (i.e.: `)`, `}`, or `]`) of the same type (i.e.: `(` matches `)`, `{` matches `}`, and `[` matches `]`). An opening brace must appear before (to the left of) its matching closing brace (e.g.: `]{}[` is *not balanced*).
- No unmatched braces lie between some pair of matched braces. For example, `({[]})` is *balanced*, but `{[}]` and `[{)]` are *not balanced*.

Create a *braces()* function. It should take an array of strings named *values* as a parameter, determine if all its braces are *balanced*, and then return an array of strings where each element indicates whether or not the element in the corresponding index of *values* was *balanced*. If the string in *values[i]* has *balanced* braces, then index *i* in the return array should contain the string `YES;` otherwise, index *i* in the return array should contain the string `NO.`

Constraints

- Max size of values parameter is 15 elements.
- Max length of strings in each element is 100.

Sample content of *values* parameter

```
["{}[]()", "{[}]}"]
```

Sample returned data from *braces()* function

```
["YES", "NO"]
```

Explanation

values[0]: `{}[]()` meets the criterion for a *balanced* string, so index *0* in our return array should contain the string `YES.`
values[1]: `{[}]` contains unmatched braces between a matched pair (in the substrings `[}`, `{[}`, and `[}]`), so index *1* in our return array should contain the string `NO`.

DELIVERABLE

A Javascript file, q7.js, containing the *braces()* function defined by the stub below:

```
/* Complete the function below. */
function braces(values) {
}
```

Q8.

You are given a pair of integers *(x,y)*. You may perform either of the two operations below, in any order, zero or more times.

1. *(x,y) -> (x+y,y)*
2. *(x,y) -> (x,y+x)*

For example, you can start with *(1, 4),* change it to *(5, 4)*, change that to *(5, 9)*, and then change that again to *(5, 14)*.
You are given four integers: *a, b, c,* and *d*. Return *Yes* if it is possible to start with the pair *(a, b)* and end with the pair *(c, d)*. Otherwise, return *No*.

Create *isPossible()* function with *4* integer parameters representing the value of *a, b ,c d* respectively. You must use **recursion** to achieve the required implementation.

Constraints

- Max value for each integer input parameter is 1000.

Output Format

Your function must return a string *Yes* if it is possible to start with the pair *(a, b)* and end with the pair *(c, d)* else return *No.*

Sample Input 1

1
4
5
9

Sample Output 1

Yes

Sample Input 2

1
2
3
6

Sample Output 2
No

Explanations
Sample Case 1:
(1, 4) can be converted to (5, 9) :
(1, 4) -> (5, 4) -> (5, 9) hence the output is *Yes*
Sample Case 2:
(1, 2) cannot be converted to *(3, 6)* hence the output is *No.*

DELIVERABLE
A Javascript file, q8.js, containing the *isPossible()* function defined by the stub below:

```
/*
 * Complete the function below.
 */
function isPossible(a, b, c, d) {

}
```

Q9.

Write a program to find the total number of *duplicate* elements in an array of size *N*. Your task is to *count* the number of elements which occur *two* or *more* times.

Create a *countDuplicates() function* which contains an integer array *numbers* as its argument. Return an integer which denotes the required result.

Constraints

- Max element in the input array is 1000.
- Max value for each integer element is 1000.

Output Format

Return an integer which denotes the required result.

Sample Input 1

```
8
1
3
1
4
5
6
3
2
```

Sample Output 1

```
2
```

Sample Input 2

```
5
1
1
2
2
3
```

Sample Output 2

2

Explanation

Sample Case 1:

N = 8

numbers = {*1, 3, 1, 4, 5, 6, 3, 2*}.

The count of duplicate elements in the array is *2*, as *1* and *3* both occur more than once.

Sample Case 2:

N = 5

numbers = {1, 1, 2, 2, 3}

The count of duplicate elements in the array is 2 as 1 and 2 both occur more than once.

DELIVERABLE

A Javascript file, q9.js, containing the *countDuplicates()* function defined by the stub below:

```
/*
 * Complete the function below.
 */
function countDuplicates(numbers) {

}
```

Q10.

Create a visual model for explaining list insertion. The model must consist of three elements:

- *List display section:* The section on the page where the list will be displayed.
- *Text field:* A field to provide input for inserting a new element into the list.
- *Button (named "Insert"):* Clicking this button inserts a new element at the end of the existing list using the value provided in the text field.

Two more requirements:

- No element should be inserted if the text field is empty when the user clicks the button.
- Every third element in the list must be displayed in *red* and the remaining elements in *black*.

DELIVERABLE

This coding exercise requires you to create three files:

- CSS file, q10.css
- Javascript file, q10.js
- HTML file, q10.html