CS 4, Winter 2007: Example Midterm Questions

This is not a practice midterm, but rather just a collection of assorted questions, from previous years, illustrative of the types of questions that will be asked. Some of these are relatively short answers, and some require a bit more, so their point values would vary. Answering these midterm questions is voluntary: do so only if you feel it is helping you with the midterm. We will not grade this.

1. What does the browser do with tags it does not understand?

2. Among the components of a URL, what is the role of the *path*?

3. What is the correct JavaScript code to write the value of the last element of array `A` in a variable named `v`?

4. What is the correct CSS rule to set the background color of paragraph of class "special" to bright green?

5. Give an example of HTML tags which indicate *physical* formatting properties.

6. What is an example of a valid type for an `<input>` tag?

7. What is the Document Object Model (DOM)?

8. Construct a valid URL from the following parts:

Protocol	http
Host	www.cs.dartmouth.edu
Path	/DigitalArts/
File	about.html

9. Write the JavaScript statement that will add 1 to the value of a variable named `examGrade`.

10. Using the BINARY SEARCH algorithm we discussed in class, the number of steps of computation required to find a value in a sorted array of n elements is proportional to what?

11. Give a precise definition of the term *algorithm*.

12. What is the purpose of the `rowspan` and `colspan` attributes of the `td` tags in HTML?

13. Use `<table>` to create a two by two checkerboard grid. Set the background color of the first row to `rgb(255,0,0)` and the text color of second row to `rgb(0,255,0)`. Use either stylesheets or inlined styles to do this. Put the numbers 1, 2, 3, and 4 in the cells in order:

1	2
3	4

14. There are (at least) four errors in the following HTML code. Circle each one and give a brief summary of what the problem is.

```
<html>
<head><title>A Document Full of Errors</title></head>
<body>
<h1>This is the CS 4 Bloopers Page</h2>
```

```
</p>I bet you probably think this web page is a candidate for
the <storng>Web Pages that Suck</strong> site!<p>
<html>
```

15. Briefly describe what overall task is accomplished by the following algorithm. Your description should specify under what circumstances the algorithm returns the value `true`, and when it returns the value `false`. For full credit, your solution must do more than simply recap the steps of the algorithm.

```
function doSomeTask(A) {
  var i = 0;
  while (i < A.length-1) {
    if(A[i] > A[i+1]) {
      return false;
    }
    i = i + 1;
  }
  return true;
}
```

16. Using the algorithm notation from the class notes, write an algorithm called `findSecondLargest` that behaves as follows: Given an array `A`, `findSecondLargest(A)` will return the second-largest value in the array. So, for example, if $A = \{3, 9, 2, 1, -8, 10, 6, 5\}$, `findSecondLargest(A)` would return 9. You may assume that `A` contains numbers.

17. Using the algorithm notation from the class notes, write an algorithm `scaleValues` which behaves as follows: Given an array `M`, `scaleValues` will find the largest and the smallest values in the array, compute their average, and then divide each element of the array by this average, replacing each element with the result of the division. The algorithm should then halt and return the average of the largest and smallest values computed in the first part.

18. Using the algorithm notation from the class notes, write an algorithm `doReverse` which behaves as follows: Given an array `A`, `doReverse` will re-arrange the elements of `A` 'in-place' (within A itself) so that they are in the reverse of their original order.

19. This HTML form is missing the Javascript function `doAddition`. Please write the function. It should load the data from the elements `alpha` and `beta`, add them together and store the results in element `gamma`. Don't worry about `<script>` tags.

```
<form name="t1">
Alpha <input type="text" name="alpha"><br>
Beta <input type="text" name="beta"><br>
Alpha+Beta <input type="text" name="gamma"><br>
<input type="button" onclick="doAddition()">
</form>
```

20. Lists vs. tables

    (a) Use **list** tags to generate the following. Don't worry about the alignment. Your answer should demonstrate that you understand how to build list from tables.

1. Alpha
   1. one
   2. two
2. Beta
   1. do
   3. re
   3. me
3. Gamma
   1. eh
   1. bee

(b) Use `<table>` tags to generate the following. Some cells will be blank.

1. Alpha	
	1. one
	2. two
2. Beta	
	1. do
	3. re
	3. me
3. Gamma	
	1. eh
	1. bee

(c) The last two questions asked you to layout the same information with two different types of tags. Describe the difference between the layouts produced by the list tag and the table tag.

21. What does the sequence `&lt;` do in HTML? Why do we need to introduce such sequences?

22. Trace the execution of the binary search algorithm given in class for the input array `A` = [1,4,6,3,5,7], when searching for the value 4. Specifically, write for each iteration the value of `low`, `mid`, `high` and `A[mid]`.

23. Consider the following HTML document:

```
<html>
<head>
  <title>COSC 4 Midterm Page</title>
</head>
<body>

<h1>Welcome to the COSC 4 Midterm Exam Home Page!</h1>

<form name="examForm">
  <p>Enter the number you got correct for each section:</p>
  <p>Part 1:  <input type="text" name="part1" size="16"> <br/>
     Part 2:  <input type="text" name="part2" size="16"> <br/>
     Part 3:  <input type="text" name="part3" size="16"></p>

  <p><input type="button" value="Click for Your Score"
            onClick="compute_score()"></p>
```

```
</form>

</body>
</html>
```

This document displays a page with a form, on which there is a push button labeled “Click for Your Score.” When this button is pressed, a JavaScript function called `compute_score()` is called. Your task is to write this function, which should behave as follows:

(a) It reads the numbers of questions answered correctly in each of the three exam sections from the three input text fields, and converts them to numbers using the built-in JavaScript function `parseInt().`

(b) It computes the final score according to the formula $s = 3 \cdot p_1 + 5 \cdot p_2 + 10 \cdot p_3$, where p_1 is the number of questions answered correctly in Part 1, p_2 is the number of questions answered correctly in Part 2, and p_3 is the number of questions answered correctly in Part 3.

(c) It displays the message ‘`Your final score is` s’, using the built-in JavaScript `alert()` function (replace s with the actual value of the final score you computed).