Assignment 4 – Arrays

Due October 9, 2013

For this assignment, please note that NO jQuery may be used. You must write everything in core JavaScript

1) Write JavaScript statements to accomplish each of the following tasks:
   a. ~~Display the value of the seventh element of array f.~~
   b. ~~Initialize each of the five elements of one-dimensional array g to 8.~~
   c. ~~Total the elements of array c, which contains 100 numeric elements.~~
   d. ~~Copy 11-element array a into the first portion of array b, which contains 34 elements.~~
   e. ~~Determine and print the smallest and largest values contained in 99-element floatingpoint array w.~~
   f. ~~Consider a two-by-three array t that will store integers.~~
   g. ~~Write a statement that declares and creates array t.~~
   h. ~~How many rows does t have?~~
   i. ~~How many columns does t have?~~
   j. ~~How many elements does t have?~~
   k. ~~Write the names of all the elements in the second row of t.~~
   l. ~~Write the names of all the elements in the third column of t.~~
   m. ~~Write a single statement that sets the elements of t in row 1 and column 2 to zero.~~
   n. Write a series of statements that initializes each element of t to zero. Do not use a repetition structure.
   o. ~~Write a nested for statement that initializes each element of t to zero.~~
   p. Write a series of statements that determines and prints the smallest value in array t.
   q. Write a statement that displays the elements of the first row of t.
   r. Write a statement that totals the elements of the fourth column of t.
   s. Write a series of statements that prints the array t in neat, tabular format. List the column subscripts as headings across the top, and list the row subscripts at the left of each row.

2) Use a one-dimensional array to solve the following problem: A company pays its salespeople on a commission basis. The salespeople receive $200 per week plus 9 percent of their gross sales for that week. For example, a salesperson who grosses $5000 in sales in a week receives $200 plus 9 percent of $5000, or a total of $650. Write a script (using an array of counters) that obtains the gross sales for each employee through an XHTML form and determines how many of the salespeople earned salaries in each of the following ranges (assume that each salesperson's salary is truncated to an integer amount):
   a) $200–299
   b) $300–399
   c) $400–499
   d) $500–599
   e) $600–699
   f) $700–799
   g) $800–899

h) $900–999
i) $1000 and over

3) Write statements that perform the following operations for a one-dimensional array:
   a. Set the 10 elements of array counts to zeros.
   b. Add 1 to each of the 15 elements of array bonus.
   c. Display the five values of array bestScores, separated by spaces.

4) Use a one-dimensional array to solve the following problem: Read in 20 numbers, each of which is between 10 and 100. As each number is read, print it only if it is not a duplicate of a number that has already been read. Provide for the "worst case," in which all 20 numbers are different. Use the smallest possible array to solve this problem.

5) Label the elements of three-by-five two-dimensional array sales to indicate the order in which they are set to zero by the following program segment:
   a. for (var row in sales)
   b. for (var col in sales[row])
   c. sales[row][col] = 0;

6) ~~Write a script to simulate the rolling of two dice. The script should use Math.random to roll the first die and again to roll the second die. The sum of the two values should then be calculated. [Note: Since each die can show an integer value from 1 to 6, the sum of the values will vary from 2 to 12, with 7 being the most frequent sum, and 2 and 12 the least frequent sums. Your program should roll the dice 36,000 times. Use a one-dimensional array to tally the numbers of times each possible sum appears. Display the results in an XHTML table. Also determine whether the totals are reasonable (e.g., there are six ways to roll a 7, so approximately 1/6 of all the rolls should be 7).]~~

7) Write a script that runs 1000 games of craps and answers the following questions:
   a. How many games are won on the first roll, second roll,…, twentieth roll and after the twentieth roll?
   b. How many games are lost on the first roll, second roll, …, twentieth roll and after the twentieth roll?
   c. What are the chances of winning at craps? [Note: You should discover that craps is one of the fairest casino games. What do you suppose this means?]
   d. What is the average length of a game of craps?
   e. Do the chances of winning improve with the length of the game?

8) A small airline has just purchased a computer for its new automated reservations system. You have been asked to program the new system. You are to write a program to assign seats on each flight of the airline's only plane (capacity: 10 seats). Your program should display the following menu of alternatives: Please type 1 for "First Class" and Please type 2 for "Economy". If the person types 1, your program should assign a seat in the first-class section (seats 1–5). If the person types 2, your program should assign a seat in the economy section (seats 6–10). Your program should print a boarding pass indicating

the person's seat number and whether it is in the first-class or economy section of the plane. Use a one-dimensional array to represent the seating chart of the plane. Initialize all the elements of the array to 0 to indicate that all the seats are empty. As each seat is assigned, set the corresponding elements of the array to 1 to indicate that the seat is no longer available. Your program should, of course, never assign a seat that has already been assigned. When the first-class section is full, your program should ask the person if it is acceptable to be placed in the economy section (and vice versa). If yes, then make the appropriate seat assignment. If no, then print the message "Next flight leaves in 3 hours."

9) Use a two-dimensional array to solve the following problem: A company has four salespeople (1 to 4) who sell five different products (1 to 5). Once a day, each salesperson passes in a slip for each different type of product actually sold. Each slip contains a) the salesperson number, b) the product number, and c) the total dollar value of the product sold that day. Thus, each salesperson passes in between zero and five sales slips per day. Assume that the information from all of the slips for last month is available. Write a script that will read all this information for last month's sales and summarize the total sales by salesperson by product. All totals should be stored in the two-dimensional array sales. After processing all the information for last month, display the results in an XHTML table format, with each of the columns representing a different salesperson and each of the rows representing a different product. Cross-total each row to get the total sales of each product for last month; cross-total each column to get the total sales by salesperson for last month. Your tabular printout should include these cross-totals to the right of the totaled rows and to the bottom of the totaled columns.

10) (The Sieve of Eratosthenes) A prime integer is an integer greater than 1 that is evenly divisible only by itself and 1. The Sieve of Eratosthenes is an algorithm for finding prime numbers. It operates as follows:

11) a) Create an array with all elements initialized to 1 (true). Array elements with prime subscripts will remain as 1. All other array elements will eventually be set to zero.

12) b) Set the first two elements to zero, since 0 and 1 are not prime. Starting with array subscript2, every time an array element is found whose value is 1, loop through the remainder of the array and set to zero every element whose subscript is a multiple of the subscript for the element with value 1. For array subscript 2, all elements beyond 2 in the array that are multiples of 2 will be set to zero (subscripts 4, 6, 8, 10, etc.); for array subscript 3, all elements beyond 3 in the array that are multiples of 3 will be set to zero (subscripts 6, 9, 12, 15, etc.); and so on. When this process is complete, the array elements that are still set to 1 indicate that the subscript is a prime number. These subscripts can then be printed.Write a script that uses an array of 1000 elements to determine and print the prime numbers between 1 and 999. Ignore element 0 of the array.

13) (Simulation: The Tortoise and the Hare) In this problem, you will re-create one of the truly great moments in history, namely the classic race of the tortoise and the hare. You will use random number generation to develop a simulation of this memorable event. Our contenders begin the race at square 1 of 70 squares. Each square represents a possible position along the race course. The finish line is at square 70. The first contender to reach or pass square 70 is rewarded with a pail of fresh carrots and lettuce. The course weaves its way up the side of a slippery mountain, so occasionally the contenders

lose ground. There is a clock that ticks once per second. With each tick of the clock, your script should adjust the position of the animals according to the rules. Use variables to keep track of the positions of the animals (i.e., position numbers are 1–70). Start each animal at position 1 (i.e., the "starting gate"). If an animal slips left before square 1, move the animal back to square 1.

Animal	Move type	Percentage of the time	Actual move
Tortoise	Fast plod	50%	3 squares to the right
	Slip	20%	6 squares to the left
	Slow plod	30%	1 square to the right
Hare	Sleep	20%	No move at all
	Big hop	20%	9 squares to the right
	Big slip	10%	12 squares to the left
	Small hop	30%	1 square to the right
	Small slip	20%	2 squares to the left

i ≤ 10. For the tortoise, perform a "fast plod" when 1 ≤ i ≤ 5, a "slip" when 6 ≤ i ≤ 7 and a "slow plod" when 8 ≤ i ≤ 10. Use a similar technique to move the hare.

Begin the race by printing
BANG !!!!!
AND THEY'RE OFF !!!!!

Then, for each tick of the clock (i.e., each repetition of a loop), print a 70-position line showing the letter T in the position of the tortoise and the letter H in the position of the hare. Occasionally, the contenders will land on the same square. In this case, the tortoise bites the hare, and your script should print OUCH!!! beginning at that position. All print positions other than the T, the H or the OUCH!!! (in case of a tie) should be blank. After each line is printed, test whether either animal has reached or passed square 70. If so, print the winner, and terminate the simulation. If the tortoise wins, print TORTOISE WINS!!! YAY!!! If the hare wins, print Hare wins. Yuck! If both animals win on the same tick of the clock, you may want to favor the turtle (the "underdog"), or you may want to print It's a tie. If neither animal wins, perform the loop again to simulate the next tick of the clock. When you are ready to run your script, assemble a group of fans to watch the race. You'll be amazed at how involved your audience gets!