Arrays, Searching & Sorting Practice Problems

# True/False Questions

1. A single array can hold multiple values of different data types.
2. Array Lists can have more than one (1) dimension.
3. A Linear Search is always more efficient than a Binary Search.
4. Arrays must be sorted for a Binary Search to work correctly.

# Multiple Choice Questions

1. Arrays are homogeneous, which means:
	1. They can only contain one type of data.
	2. They can contain multiple types of data.
	3. They are static in size.
	4. The index starts at 1.
2. To traverse multidimensional arrays, you most commonly use \_\_\_\_\_ loops:
	1. NOT
	2. pointer
	3. infinite
	4. nested
3. Which of these sorts was NOT covered in the CSE 1321 slides:
	1. Bubble
	2. Insertion
	3. Selection
	4. Merge
4. 1D & 2D arrays are:
	1. Used to store multiple types of data
	2. Used to prevent data overflow errors
	3. Used to store multiple values of the same data type
	4. Used to confuse students

# Programming & Problem-Solving Questions – 1D Arrays

1. Output from traversing an array

BEGIN MAIN

 CREATE multiplier = 1;

 CREATE[] myArray = {1,1,1,1,1}

 FOR (CREATE i = 0, i < length of myArray, i++)

 multiplier \*= 2;

 myArray[i] += multiplier

 END FOR

 FOR (CREATE p = 0, p < length of myArray, p++)

 PRINT (myArray[p] + “, “)

 END FOR

END MAIN

* 1. What is the final value of multiplier?
	2. What is the exact output of the second FOR loop?
1. Imagine we have a 1D array (called “bob”) of 10,000 random numbers that will be fed into a program. However, if there are three zeroes in a row (000), the program will break, the computer will smoke, and life as we know it will cease to exist. Write code that scans array *bob* and detects if three zeroes appear in a row. If three zeroes do appear in a row, print “SMOKE! RUN!”. Otherwise, the program should not output anything. Hint: start by finding the first zero. What should you check for next?
2. Exponential growth is amazing, especially when it comes to investing! For this question, you’re going to create an array that holds numbers (representing dollars). Then you’re going to initialize each slot (or cell) of the array with the previous cell’s value multiplied by 1.08 (or 8% growth). Hint: this is nearly identical to initializing an array the regular way (with 1’s, for example)

Declare an array of 500 numbers and initialize it per the instructions above. To do this, you’ll want to put the value 1 in the first slot/cell. For the remaining, you can calculate a cell’s value based on the previous cell’s value.

1. Did you know that unlucky numbers vary around the world? For example, China’s is 4, Japan’s is 9, Italy’s is 17, and Afghanistan’s is 39. For this question, you will do two things. First, in pseudocode, you will declare an array of 1,000,000 numbers. Second, we’re going to help Italy out and scan the array to make sure the number 17 doesn’t appear in the array. Before you scan it, you should initialize the array with random numbers. Using the array above, write code to scan the array – searching for the number 17. After scanning, if a 17 was found, print “Che brutto!” (How horrible!). If 17 isn’t found, print “Buona fortuna!” (Good luck!).
2. Create an array of size 100 to store student grades called StudentGrades. Fill it with random grades from 0 to 100. Then complete the following updates:
	1. The value in the 72nd cell/slot is incorrect!  Write the code to change its value to 86.
	2. The value in the 37th cell/slot is incorrect!  Write the code to change its value to 93.
	3. The value in the 44th cell/slot is incorrect!  Write the code to change its value to 42.
	4. The value in the 55th cell/slot is incorrect!  Write the code to change its value to 68.
	5. The value in the 100th cell/slot is incorrect!  Write the code to change its value to 57.
	6. The value in the 1st cell/slot is incorrect!  Write the code to change its value to 71.
	7. Write the code to sum the elements stored in the odd indices in the array of grades. After processing the array, print the sum with a descriptive message.
	8. Write the code to sum the even indices in the array of grades. After processing the array print the average of the even numbers with a descriptive message.
	9. Write the code to find the largest element in the array of grades. After the array has been processed, print the index of the largest element found with a descriptive message.
	10. Write the code to find the smallest element in the array of grades. After the array has been processed, print the value that was found and its index with a descriptive message.
	11. Write the code to increase the value of each element in the array by 10. After the array has been processed print “A 10-point curve has been applied.” to let the user know the program is done.
	12. Write the code to square each element in the array of grades. After the array has been processed, print the “Array has been squared.” to let the user know it is done.
3. Declare and initialize a 1D array of 2300 Booleans to store whether or not a group of 2300 people have seen the latest movie in the Marvel Cinematic Universe in a movie theater.
4. Declare and initialize a 1D array of 3300 Bytes to store the ages of the players and coaches involved in Major League Baseball.
5. Declare and initialize a 1D array of 3,400 Characters to store the Final Grades of the students who graduated from KSU last semester.
6. Declare and initialize a 1D array of 35,600 Doubles to store the GPAs of all the students enrolled in KSU.
7. Declare and initialize a 1D array of 1,138 Strings to store the names of the students in the FYE program.
8. Create an array of Strings (size is 50) called PossibleLocations. Initialize every slot to null then
	1. The value in the 11th cell/slot is incorrect!  Write the code to change its value to “Atlanta”.
	2. The value in the 21st cell/slot is incorrect!  Write the code to change its value to “Portland”.
	3. Write the code to print each element in the array starting from the end of the array, with a new line after every five elements. After the array has been completely printed, print the “Array has been printed.” to let the user know it is done.
	4. Write the code to print each element in the array starting with the beginning of the array, with a new line after every five elements. After the array has been completely printed, print the “Array has been printed.” to let the user know it is done.
9. write a complete program that declares a 1D array of size ten named anArray. Using a WHILE loop to traverse the array you must fill the array with names input by the user, the PRINT statement should indicate which index is currently being filled (E.g.: “Please enter element for index 0: “). Write a method that traverses the 1D array using a FOR loop to traverse the array to determine if there are any duplicate names stored adjacent to each other in the array and print the duplicate names (if any) AFTER the array is fully traversed.
10. Write a complete program that declares a 1D array of size one hundred named anArray. Using a WHILE loop to traverse the array you must fill the array with numbers input by the user, the PRINT statement should indicate which index is currently being filled (E.g.: “Please enter element for index 0: “). Write a method that traverses a 1D array using a FOR loop to traverse to find the largest value input by the user and print it AFTER the array is fully traversed.
11. Write a complete program that declares a 1D array of size forty-two named anArray. Using a WHILE loop to traverse the array you must fill the array with characters input by the user, the PRINT statement should indicate which index is currently being filled (E.g.: “Please enter element for index 0: “). Write a method that uses a FOR loop to traverse the array to count the number of hashtags (#) input by the user and print the total AFTER the array is fully traversed.
12. Create an array of 100 bank account balances called AccountBalances. Fill it with random values between 0.00 and 1,000,000.00, then:
	1. The value in the 1st slot is incorrect!  Change its value to $77,500.
	2. The value in the 100th slot is incorrect!  Change its value to $56,792.
	3. The value in the 27th slot is incorrect!  Change its value to $23,542.
	4. The value in the 90th slot is incorrect!  Change its value to $27,312
	5. The value in the 21st slot is incorrect!  Change its value to $97,123
	6. The value in the 59th slot is incorrect!  Change its value to $57,545
	7. The value in the 79th slot is incorrect!  Change its value to $37,500.
	8. Print the index of all the accounts that have a negative balance
	9. Print the index of all the accounts that have a balance equal to $125,000
	10. Print the index of all the accounts that have a balance greater than $100,000
	11. Print the index of all the accounts that have a negative balance
	12. Print the index of all the accounts that have a $0 balance
	13. Print the index of all the accounts that have $30,000 or more in them
	14. Print the index of all the accounts that contain less than $30,000 in them
13. Write the code that declares an array of INTEGERS, called myArray, with a size of 100 elements and initializes the array. Your code must use a FOR loop, prompt the user to enter numbers that are in the range of 0 to 255, and store the numbers in the array. If the entered numbers are within the proper range, they are stored in the array. Otherwise, an error message is printed, and another number should be entered.
14. Write the code that searches the array from # 16 for the first value that is larger than a target value. Your code must prompt the user to enter a target value and must use a WHILE loop to find the first value in the array that is larger than the target value entered by the user. If such a value is found, the code should print the message, “The first larger value found is 23”, and immediately end the loop. Otherwise, the search should continue until the entire array has been searched. Finally, after the WHILE loop ends, your code should print the statement: “The while loop has ended.”

For example, the user enters a target value of 14. For an array with the following values: 10, 3, 71, 11, 75, 66, 16, the message printed should be: “The first larger value found is 71”, then the message printed after the loop ends should be: “The while loop has ended.”

1. Write the code that declares an array of DOUBLES, called myArray, with a size of 100 elements. And initializes the array. Your code must use a FOR loop, prompt (ask) the user to enter numbers that are in the range of 0.0 to 4.0 and store the numbers in the array. If the entered numbers are within the proper range, they are stored in the array. Otherwise, an error message is printed, and another number should be entered.
2. Write the code that searches the array from # 18 for a target value. Your code must prompt the user to enter a target value and must use a WHILE loop. Once the target value is found, the code should print the message, “Target value found”, and immediately end the loop. Otherwise, the search should continue until the entire array has been searched. Finally, after the WHILE loop ends, your code should print the statement: “The while loop has ended.”

For example, the user enters a target value of 14. For an array with the following values: 10, 3, 14, 11, 75, 66, 16, the message printed should be: “Target value found”, then the message printed after the loop ends should be: “The while loop has ended.”

1. Write the code that declares an array of CHARACTERS, called myArray with a size of 25 elementsand initializes the array. Your code must use a FOR loop, prompt (ask) the user to enter characters that are in the range of “A” to “Z” (use upper case letters only) and store the characters in the array. If the entered characters are within the proper range, they are stored in the array. Otherwise, an error message is printed, and another character should be entered.
2. Write the code that searches the array from # 20 for a target value, and counts the number of times the target value is found. Your code must prompt the user to enter a target value and must use a WHILE loop. Finally, after the WHILE loop ends, your code should print a message indicating the number of times the target value was found, i.e. “The target value was found 3 times.”

For example, the user enters a target value of ‘k’. For an array with the following values: ‘Z’, ‘k’, ‘9’, ‘L’, ‘&’, ‘k’, the message printed after the loop ends should be: “The target value was found 2 times.”

1. Write the code that declares an array of BOOLEANS, called myArray, with a size of 50 elements and initializes the array. Your code must use a FOR loop, prompt the user to enter true or false values (i.e. “T” or “t”), and store the values in the array. If the user enters “T” or “t” the value stored in the array is true. Otherwise, the value stored in the array is false.
2. Write the code that searches the array from #22 for a target value, and counts the number of times the target value is found. Your code must prompt the user to enter a target value (either a true or false value), and must use a WHILE loop. Finally, after the WHILE loop ends, your code should print a message indicating the number of times the target value was found, i.e. “The target value was found 3 times.”

For example, the user enters a target value of true. For an array with the following values: true, true, false, true, false, false, false, the message printed after the loop ends should be: “The target value was found 3 times.”

1. During an interview with the TVGuide Channel, you have been tasked with creating an array named TVChannelNames. You should create it and initialize all cells to null then
	1. Update cell/slot #87 in the array named TVChannelNames to “The CW”.
	2. Update cell/slot #45 in the array named TVChannelNames to “Freeform”.
	3. Update cell/slot #12 in the array named TVChannelNames to “Paramount Network”.
	4. Print all elements that are even
	5. Printing all elements from indices 10 to 19
	6. Print all elements that are divisible by 5

# Programming & Problem-Solving Questions – 2D Arrays

1. Create a 4 by 6 2D array of integers and populate it with random numbers between 0 and 100 using a random number generator. Utilizing the 2D array you created in the previous question, use the search algorithm of your choice to see if 42 appears in the array. Find all occurrences of 42 within the array and multiply them by two. If it does not appear print “Sorry, that value does not exist.”
2. Create a 7 by 5 2D array of shorts and populate it with random numbers between -50 and 50 using a random number generator. Utilizing the 2D array you created in the previous question, use the search algorithm of your choice to see if 13 appears in the array. Find all occurrences of 13 within the array and change the value to zero. If it does not appear print “Sorry, that value does not exist.”
3. Create a 6 by 4 2D array of longs and populate it with random numbers between 25 and 125 using a random number generator. Utilizing the 2D array you created in the previous question, use the search algorithm of your choice to see if 72 appears in the array. Find all occurrences of 72 within the array and add 10 to each one. If it does not appear print “Sorry, that value does not exist.”
4. Create a 5 by 3 2D array of bytes and populate it with random numbers between -100 and 0 using a random number generator. Utilizing the 2D array you created in the previous question, use the search algorithm of your choice to see if -88 appears in the array. Print the number of times -88 appears in the array. If it does not appear print “Sorry, that value does not exist.”
5. Imagine you have a 2D array of Characters named CHECKOFF containing either ‘X’ or null in each cell. Your boss has asked you to traverse the array and determine if any rows are empty (all null values). If a row is empty, return true, else return false once you have traversed the entire array.
6. Imagine you have a 2D array of Characters named UPPERCASE containing an uppercase character in each cell. Your boss has asked you to traverse the array and determine if any rows contain both ‘X’ and ‘R’. If a row does contain both, return true, else return false once you have traversed the entire array. HINT: Keep track of whether you have seen an ‘X’ or ‘R’ as you traverse each row.
7. Imagine you have a 2D array named WORDS, your boss has asked you to write the code that would traverse the diagonals of the array to concatenate the elements into a single String of text (e.g. [0,0], [1,1]) and print the resulting String.
8. Imagine you have a 2D array of Integers named DATA, your boss has asked you to write the code that would traverse the entire array and print out the largest number from each row of the array.
9. Imagine you have a 2D array of Integers named DATA, your boss has asked you to write the code that would traverse the entire array and print out the sums of each column of the array.
10. A DNA testing company has given you a 10x10 array of characters called DNA which contains a sequence of DNA bases (letters G, T, A, and C). They need you to process the array, so that it counts the number of each base and prints the results out to the console for the researchers to use.
11. The IRS has contracted you to process a 10x10 array of floating-point numbers called *Taxes* and sum up all negative numbers in the array so they know how much money they need to collect from delinquent taxpayers.
12. Scientific Games® has given you a 10x10 array of floats called *Data* which is supposed to contain values from -1.0 to +1.0; however, there might be errors! They need you to process the array so that it verifies all elements in the array fall within the specified range of -1.0 to +1.0, printing “Valid” or “Invalid”.
13. A mathematician has given you a 10x10 array of Boolean values called *Attending* which contains the results of their poll to see whether students have interest in attending an upcoming conference. You are to process the array so that it counts the number of each result and prints which value (true or false) occurs the most, so they know whether or not to hold the conference.
14. You have been given a 10x10 array of integers, called Puzzle. You are to process the array, so that it verifies that each column’s elements are equal to the column’s index. That is, all elements in column 0 should be a 0, all elements in column 1 should be a 1, and so on.
15. Imagine you have been given an array named myArray, it contains random integers between 0 and 1000. Write code that processes a 2D array of integers and prints the number of times the value 42 appears in the array.
16. Imagine you have been given an array named myArray, it contains random doubles between 0.0 and 4.0. Write code that processes a 2D array of doubles and prints the number of elements that are less than 2.0 in the array.
17. Imagine you have been given an array named myArray, it contains random Booleans (true/false). Write code that process a 2D array of Booleans and prints the number of false elements in the array.
18. Imagine you have been given an array named myArray, it contains random characters, all lowercase English alphabet letters. Write code that processes a 2D array of characters and prints the number of non-vowels (consonants: b, c, d, etc.) in the array.
19. Write a program that traverses a 2D array of floats and finds the minimum and maximum values in the array and swaps them.
20. Write a program that traverses a 2D array of floats and calculates the average of the of the array and replaces all instances of the average with -1.
21. Write a program that traverses 2D array of floats to locate a target value, counting number of times the target value appears and printing a statement to show how many times it was found.
22. Write a program that uses a 5 X 5 two-dimensional array of doubles called creditCardBalances.
	1. Each row in the array represents a customer, and each column in the array represents the customer’s credit card balance for the five different credit cards. So, there are five customers, and each customer has five credit cards.
	2. **For each customer**, the program calculates the total balance of all five credit cards and then prints the row number and balance for only the customers who have a balance over 10,000.00.
	3. **For each customer**, the program identifies the bank account with the largest balance and then prints the row number and its largest balance.
	4. **For each customer**, the program sums the balance on all credit cards that have a balance over 10,000.00 and then prints the row number and sum.