2015 ap computer science free response answer

Creating a full class has been a popular free answer question in recent years in the AP Computer Science experiment. In 2015, they asked you to create a category called HiddenWord that imitated the word guessing game. The idea was that the computer knew the word and you tried to guess it. The computer would then take your guess and compare it to the answer. If you had the same letter in the same position, the letter was returned to the tip. If your guess was a corresponding letter, but in the wrong place, the tip would have + in that position. If there's no letter, * would be in that position. For example, assume that the answer message was HELLO. Here are a few guesses and what they would return. ABCDF would return *****. There are no such letters. HOWDY would return the H****. It's got an H, and it's in the right position. They'd come back *++**. It's H and O, but they're in the wrong position. HELPS would return the HEL**. HELLO would return HELLO Full category Here's the work solution I came up with. public category HiddenWord { private string word; public HiddenWord(String w) { word = w; } public string getHint(String guess) { String out = ; for (int i=0; i&lt;guess.length(); i++)= {= string= guesscharacter=guess.substring(i, i+1);= string= wordcharacter=word.substring(i, i+1);= if= (guesscharacter.equals(wordcharacter))= {= out= +=guessCharacter; } = else= if= (word.indexof(guesscharacter)=&gt;= 0) { out += +; } else { out += *; } return out; } } You must know that you can use HiddenWord in a public category to define a category called hiddenword. For an AP exam, classes are public. An instance variable is required to keep the response and must be a string. I named my word. It should be private. The problem gave an invitation from the sample constructor to forward the response string. It showed something like this - HiddenWord game = new HiddenWord (HELLO);. It tells us that our constructor must take exactly one String parameter. And you need to enable a method called getHint, which also takes up the string parameter. then the guess would be compared to a word that looks at every sign. If the signs match, the character it's put out. If they don't match, but the guess sign is in a word, the + symbol is put out. And if neither is true, then * will be put out. Want to stay in touch and stay up to date with the latest posts @ CompSci.rocks? Chapter 6 Notes Solutions to Previous Free Questions This material will be provided to you, supplemented by a book prepared for the AP computer science exam in Java. You do not have the authority to publish or distribute it in any form without our permission. However, you can print one copy of this chapter and files for personal use and face-to-face teaching for each copy of the Book of Prepared book you own or received from your school. This chapter provides solutions and notes for free answer questions for exams 2004–2019. &lt;/guess.length();&gt;questions and evaluation guidelines for these exams are published apstudent.collegeboard.org and teachers at AP Central: The best way to practice previous exam questions is first to solve them yourself and then compare our solutions with our solutions and notes. Today, the 2015 APCS FRQs were released. Here are my solutions. I created all the extra code to test all these categories and interfaces as well. But these are just answers to questions. For a BlueJ project with categories for testing code, click here. If you just want to browse the categories, click here. Question 1 //part Public static int arraySum(int[] arr) { int sum = 0; for(int n: arr) sum += n; return amount; } //part B public static int[] line Amounts(int[][] arr2D) { int[] amounts = new int[arr2D.length]; for(int i = 0; i &lt; sums.length; i++) { sums[i] = arraySum(arr2D[i]); } return amounts; } //part C public static boolean isDiverse(int[][] arr2D) { int[] amounts = rowSums(arr2D); for(int i = 0; i &lt; sums.length; i++) for(int j = i+1; j &lt; sums.length; j++) if(sums[i] == sums[j]) returns false; returns true; } Question 2 public category HiddenWord { private string hidden; public HiddenWord(String h) { hidden = h; } public string getHint(String hint) { String r = ; for(int i = 0; i &lt; hint.length(); i++) { if(hint.charAt(i) == hidden.charAt(i)) r += +hint.charAt(i); otherwise if(hidden.indexOf(hint.charAt(i)) &gt; -1) r += += +; other r += *; } return r; } } } Question 3 // Part Public int getValueAt(int row, int col) { for(SparseArrayEntry e: entries) { if(e.getRow() == line &amp;&amp; e.getCol() == col) returns e.getValue(); } returns 0; } / Part B public vacuum removesColumn(int col) { numCols---; for(int i = sizetries.()-1; i &gt;= 0; i--) if(entries.get(i).getCol() == col)tries.remove(i); for(int i = 0; i &lt; entries.size(); i++) if(entries.get(i).getCol() &gt;= col) { SparseArrayEntry h = entries.get(i); SparseArrayEntry e = new SparseArrayEntry(h.getRow(),(h.getCol()-1),h.getValue()); markings.set(i, e); } } Question 4 // part Public interface NumberGroup { public boolean value contains(int num); } // part B public category The area implements the NumberGroup { private int[] list; public area(int min, int max) { list = new int[Math.abs(max-min+1)]; for(int i = 0; i &lt; list.length; i++) list[i] = min + i; } public boolean contains(int num) { for(int n: list) if(num == n) return true; return false; } } // part C public boolean contains(int num) { for(NumberGroup n: groupList) if(n.contains(num)) return true; return false; } DiverseArray free answer problem 2015 AP Computer Science A Exam. DiverseArray #1 about the 2015 AP Computer Science A Free Response issues. Part (a) — arraySum public static int arraySum(int[] arr) { int sum = 0; for(int value: arr) sum += return sum; } Part (b) – rowSums public static int[] rowSums(int[][] arr2D) { int[] amounts = new new for(int i = 0; i &lt; sums.length; i++) amounts[i] = arraySum(arr2D[i]); return amounts; } This method requires an invitation to arraySum, so the 2D table arr2D must be treated as a 1D array matrix. A regular loop is suitable here, since the index is needed for both amounts and arr2D. Part (c) – isDiverse method public static boolean isDiverse(int[][] arr2D) { int[] sums = rowSums(arr2D); for(int i = 0; i &lt; sums.length; i++) for(int j = i + 1; j &lt; sums.length; j++) if(sums[i] == [j]) returns false; returns true; } I'm not usually an advocate for nested loops of 1D matrix pass-through. However, they are a simple solution here. An alternative solution is to sort amounts (not arr2D), and then check for adjacent duplicates. The map can also be used. However, this does not fall within the scope of AP CS A.A.