2018 AP Comp Science A FRQ

1. Parts A and B
   1. simulate()
      1. Didn’t get a copy of the code
   2. runSimulation()

| public double runSimulation(int num){  double gudRuns = 0;  double totalRuns = (double)num;  for(int i = 0; i < num; i++){  boolean run = simulate();  if(run){  gudRuns++;  }  }  return(gudRuns / totalRuns);  } |
| --- |

1. Parts A and B
   1. wordPairList

| public wordPairList(String[] words){  for(int i = 0; i < words.length - 1; i++){  for(int q = i + 1; q < words.length; q++){  allPairs.add(new WordPair(words[i], words[q]));  }  }  } |
| --- |

* 1. numMatches

| public int numMatches(){  int count = 0;  for(wordPair p : this.allPairs){  if(p.getFirst().equals(p.getSecond())){  count++  } }  return count;  } |
| --- |

1. String Checker
   1. codeWordChecker

| public class codeWordChecker implements StringChecker{  private int min;  private int max;  private String bad;  public codeWordChecker(String bad){  this.min = 6;  this.max = 20;  this.bad = bad;  }  public codeWordChecker(int min, int max, String bad){  this.min = min;  this.max = max;  this.bad = bad;  }  public boolean isValid(String str){  return str.length()>= this.min && str.length() <= this.max && str.indexOf(this.bad) == -1; |
| --- |

1. Latin Squares (Matrix)
   1. getColumn

| public static int[] getColumn(int [][] arr2D, int c){  int[][] final = int[arr2D.length][arr2D[c].length];  for(int i = 0; i < arr2D.length; i++){  for(int q =0; q < arr2D[i].length; q++){  final[i][q] = arr2D[i][c];  }  }  return final;  } |
| --- |

* 1. isLatin

| public static boolean isLatin(int [][] square){  boolean iNeedToBreak = false;  Array nums = new Array[square.length]();  for(int i = 0; i < nums.length; i++){ nums[i] = square[i][0];  }  while(i <= nums.length){  for(int q = 0; q <= nums.length; q++){  if(nums[i].equals(nums[q])){  nums[q] = null;  iNeedToBreak = true;  }  }  if(iNeedToBreak){  return false;  }  int iHaveIt = 0;  for(int r = 0; r < square.length; r++){  for(int c = 0; c < square[r].length; c++){  for(int count = 0; count < nums.length; count++){  if(square[r][c].equals(nums[count])){  iHaveIt++;  }  }  }  }  if(iHaveIt == nums.length){  return true;  }  else{  return false;  } |
| --- |