2016 FRQ Responses

Name: Shreyash Ranjan

**1a. Write the entire RandomStringChooser class.**

Public class RandomStringChooser {

Private List<String> words;

Public RandomStringChooser(String[] wordArr)

Words = new ArrayList<String>();

For (String singleWord : wordArr) {

words.add(singleWord);

}

}

Public String getNext() {

If (words.size() >0) {

Return words.remove ((int) (Math. random()\*words.size()));

}

Return “NONE”;

}

}

**1b.**

public RandomLetterChooser ( String str ){

Super (getSingleLetters(str));

}

**2a.**

public LogMessage ( String message ){

Int colon = message.indexOf(“:”);

machineId = message.substring(0, colon);

Description = message.substring(colon +1);

}

**2b.**

**public boolean containsWord ( String keyword ){**

If (description.equals(keyword()) {

Return true;

}

If (description.indexOf(keyword + “ “) == 0) {

Return true;

}

If (description.indexOf(“ “ + keyword() != -1) {

Return true;

}

If (description.length() > keyboard.length()) {

If ((description.substring(description.length()-keyword.length()-1.equals( “ “ + keyboard)))

}

Return true;

}

}

Return false;

}

**2c.**

**public List<LogMessage> removeMessages ( String keyword ){**

List <LogMessage> removals = new ArrayList<LogMessage>();

For (int i=0; i<messageList.size(); i++) {

If (messageList.get(i).containsWord(keyword)) {

removals.add(messageList.remove(i));

I--;

]

}

Return removals;

}

**3a.**

private boolean toBeLabeled ( int r, int c, boolean[ ][ ] blackSquares ){

Return (!(blackSquares[r][c]) && (r == 0 || c ==0 || blackSquares [r-1][c] || blackSquare[r][c-1]));

}

**3b.**

public Crossword ( boolean [ ][ ] blackSquares ){

Puzzle = new Square[blackSquares.length][blackSquares[0].length];

Int num = 1;

For ( int r = 0; r < blackSquares.length; r++) {

If (blackSquares [r][c]) {

Puzzle[r][c] = new Square (true, 0);

}

Else {

If (toBeLabeled(r, c, blackSquyares)) {

Puzzle[r][c] = new Square(false, num);

Num++;

}

Else {

Puzzle[r][c] = new Square(false,0);

}

}

}

}

}

**4a.**

**public static int totalLetters ( List<String> wordList ){**

Int total = 0;

For (String word : wordList) {

Total += word.length();

}

Return total;

}

**4b.**

**public static int basicGapWidth ( List<String> wordList, int formattedLen ){**

Return (formattedLen - totalLetters(wordList)) / (wordList.size()-1;

}

**4c.**

**public static String format ( List<String> wordList, int formattedLen ){**

String frommated = “”;

Int gapWidth = basicGapWidth(wordList, forattedLen);

Int leftovers = leftoverSpaces(wordList, formattedLen);

For (int w = 0; w<wordList.size() - 1; w++) {

formatted= formatted + “ “;

}

If (leftovers > 0) {

Formatted = formatted + “ “;

leftovers --

}

}

Formatted = formatted + wordList.get(wordList.size()-1);

Return formatted;

}