Question 1

(a) Part

```
public int scoreGuess(String guess) {
    int i = 0;
    int count = 0;
    int findIndex = secret.substring(i).indexOf(guess);

    while (findIndex != -1) {
        findIndex = secret.substring(i).indexOf(guess);

        if (findIndex != -1) {
            count++;
            i = i + findIndex + 1;
        } else {
            break;
        }
    }

    int finalScore = count * guess.length() * guess.length();
    return finalScore;
}
```

(b) Part

```
public String findBetterGuess(String guess1, String guess2) {
    int guess1score = scoreguess(guess1);
    int guess2score = scoreguess(guess2);

    if (guess1score > guess2score) {
        return guess1;
    }

    if (guess1score < guess2score) {
        return guess2;
    }

    if (guess1.compareTo(guess2) > 0) {
        return guess1;
    }

    if (guess1.compareTo(guess2) < 0) {
        return guess2;
    }
}
```

Question 2

```
public class CombinedTable {
    private SingleTable table1;
    private SingleTable table2;

    public CombinedTable(SingleTable t1, SingleTable t2) {
        table1 = t1;
        table2 = t2;
    }
    public boolean canSeat(int num) {
        if (num > (table1.getNumSeats() + table2.getNumSeats()) - 2) {
            return false;
        } else {
            return true;
        }
    }

    public double getDesirability() {
        if (table1.getHeight() == table2.getHeight()) {
            return ((table1.getViewQuality() + table2.getViewQualit()) / 2);
        } else {
            return ((table1.getViewQuality() + table2.getViewQuality()) / 2) - 10;
        }
    }
}
```

Question 3

(a) Part

```
public void addMembers(String[] names, int gradYear) {
    for (String name : names) {
        MemberInfo temp = new MemberInfo(name, gradYear, true)
        memberList.add(temp);
    }
}
```

(b) Part

```
public ArrayList<MemberInfo> removeMembers(int year) {
    ArrayList<MemberInfo> returnList = new ArrayList<MemberInfo>();

    for (int index = memberList.size() - 1; index >= 0; index--) {
        MemberInfo current = memberList.get(index);

        if (current.getGradYear() <= year) {
            if (current.inGoodStanding()) {
                returnList.add(current);
            }

            memberList.remove(index);
        }
    }

    return returnList;

}
```

Question 4

(a) Part

```
public static boolean isNonZeroRow(int[][] array2D, int r) {
    for (int index = 0; index < array2D[r].length; index++) {
        if (array2D[r][index] == 0) {
            return false;
        }
    }

    return true;
}
```

(b) Part

```
public static int[][] resize(int[][] array2D) {

    int[][] result = new int[numNonZeroRows(array2D)][array2D[0].length];

    int newRowIndex = 0;

    for(int[] row : array2D) {
        if (isNonZeroRow(array2D, row)) {
            for (int col: row) {
                result[newRowIndex][col] = array2D[row][col];
            }
            newRowIndex++;
        }
    }

    return result;
}
```