CSA FRQ 2023

Question 1

```
public int findFreeBlock(int period, int duration){
  boolean isFree = false;
  for(int i = 0; i <= 60 - duration; i++){
    isFree = true;
    for(int j = i; j <= i + duration - 1){
      if(isMinuteFree(period, j) == false){
        isFree = false;
        break;
      }
    }
    if(isFree){
      return i;
    }
  }
  return -1;
}
```

```
public boolean makeAppointment(int startPeriod, int endPeriod, int duration){
  for(int i = startPeriod; i <= endPeriod; i++){
    int minuteFree = findFreeBlock(i, duration);
    if( minuteFree != -1){
      reserveBlock(i, minuteFree, duration);
      return true;
    }
  }
  return false;
}
```

Question 2

```
class Sign{
  private String message;
  private int width;
  public sign(String message, int width){
    this.message = message;
    this.width = width;
  }
  public int numberOfLines(){
    int len = message.length();
    if (len == 0){
```

```
      return 0;
    }
    if(len % width == 0){
      return len/width;
    } else{
      return (len/width) + 1;
    }
  }
  public String getLines(){
    String lineForm = "";
    int startPosition = 0;
    if(message.length() == 0){
      return null;
    }
    for(int i = 1; i <= numberOfLines(); i++){
      if(i == numberOfLines()){
        lineForm = lineForm + message.substring(startPosition);
        break;
      }
     lineForm = lineForm + message.substring(startPosition, i * width);
     startPosition += width;
     lineForm = lineForm + ";";
    }
    return lineForm;
  }
}
```

Question 3

```
public void cleanData(double lower, double uppper){
  for(int i = temperatures.size() - 1; i >= 0; i--){
    if(temperatures.get(i) > upper || temperatures.get(i) < lower){
      temperatures.remove(i);
    }
  }
}
```

```
public int longestHeatWave(double threshold){
  int waveLength = 0;
  int count = 0;
  for(int i = 0; i < temperatures.size(); i++){
    if(teperatures.get(i) > theshold){
      count ++;
    } else{
      if(count > waveLength){
        waveLength = count;
        count = 0;
```

```
      } else{
        count = 0;
      }
    }
  }
  if (count > waveLength) {
        waveLength = count;
  }
  return waveLength;
}
```

Question 4

```
public boolean moveCandyToFirstRow(int col){
  if(box[o][col] != null){
    return true;
  }
  for(int row  = 1; row < box.length; row++){
    if(box[row][col] != null){
      box[0][col] = box[row][col];
      box[row][col] = null;
      return true;
    }
  }
  return false;
}
```

```
public Candy removeNextByFlavor(String flavor){
  Candy temp = null;
  String tempFlavor = null;
  for(int row  = box.length - 1; row >= 0; row--){
    for(int col = 0; col < box[row].length; col++){
      if(box[row][col] == null){
        continue;
      }
      temp = box[row][col];
      tempFlavor = temp.getFlavor();
      if(tempFlavor.equals(falvor)){
        box[row][col] = null;
        return temp;
      }
    }
  }
  return null;
}
```