1. Parts A and B
   1. A

| public static int numberOfLeapYears(int years1, int year2) {  int count = 0;  for(int i = 0; i <= year2; i++) {  if(isLeapYear(i)) {  count++;  }  }  return count;  } |
| --- |

* 1. B

| public static int dayOfWeek(int month, int day, int year) {  Int count = dayOfYear(month, day, year);  count+= firstDayOfYear(year);  count--;  count = count % 7;  return count;  } |
| --- |

1. Part A
   1. A

| class StepTracker {  private int active;  private int totalSteps;  private final int goal;  private int days;  public stepTracker(int goal) {  this.goal = goal;  totalSteps = 0;  days = 0;  active = 0;  }  public void addDailySteps (int steps) {  this.totalSteps += steps;  if(steps >= 10000) {  this.active++  }  days++;  }  public int activeDays() {  return active;  }  public double averageSteps() {  double end = (double) (totalSteps/days);  return end;  }  } |
| --- |

1. Part A and B
   1. A

| public ArrayList<String> getDelimitersList(String[] tokens) {  ArrayList<String> output = new ArrayList<String>();  for(String s : tokens) {  if(s.equals(openDel) || s.equals(closeDel)) {  output.add(s);  }  return output;  } |
| --- |

* 1. B

| public boolean isBalanced(ArrayList<String> delimiters) {  int openCount = 0;  int closeCount = 0;  for(String s: delimiters) {  if(s.equals(openDel)) {  openCount++;  }  else {  closeCount++;  }  if(closeCount > openCount) {  return false;  }  }  return openCount == closeCount;  } |
| --- |

1. Parts A and B
   1. A

| public LightBoard(int numRows, int numCols) {  Light = new boolean[numRows][numCols];  for(int i = 0; i < numRows; i++) {  for(int 1 = 0; q < numCols; q++) {  if((int)(Math.random \* 10) < 4 ) {  lights[i][q] = true;  }  else{  light[i][q] = false;  }  }  }  } |
| --- |

* 1. B

| public boolean evalutateLight(int row, int col) {  int colCount = 0;  for(int i = 0 ; i < lights.length; i++) {  if(lights[i][col]){  colCount++;  }  }  if(lights[row][col] && colCount % 2 == 0) {  return false;  }  if(!lights[row][col] && colCount % 3 == 0) {  return true;  }  return lights[row][col];  } |
| --- |