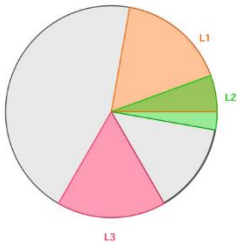
DSA

Name ……. ID ……

You are given n light beams in a circular room. Each light beam is described by two angles: a starting angle and an ending angle, both in degrees between 0 and 360.

A beam lights up the arc from the starting angle to the ending angle (inclusive) in a clockwise direction. Multiple beams may overlap if their lighting ranges intersect.

Your task is to merge all overlapping beams and return the minimum number of non-overlapping beams that together cover the same total area. The example input is



Input: n and 2 * n values for start, end representing a beam as [start, end]

3

10 90 70 100 150 210

Output: A list of merged beams, each also represented as [start, end]

[10, 100]

[150, 210]

```
static void demo_1() {
    int n = 3;
    int [] data = {10, 90, 70, 100, 150, 210};
    List<int[]> beams = new ArrayList<>();
    for (int i = 0; i < 2*n; i+=2) {
        beams.add(new int[] {data[i],data[i+1]});
    }
    List<int[]> merged = mergeBeams(beams);

    for (int[] beam : merged) {
        System.out.println(Arrays.toString(beam));
    }
}
```