

## Fill Grid

Given two one-dimensional arrays LL and UR, fill in the program on the next page to insert the elements of LL into the lower-left triangle of a square two-dimensional array S and UR into the upper-right triangle of S, without modifying elements along the main diagonal of S. You can assume LL and UR both contain at least enough elements to fill their respective triangles. (Spring 2020 MT1)

For example, consider

```
int[] LL = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 0, 0 };
int[] UR = { 11, 12, 13, 14, 15, 16, 17, 18, 19, 20 };
int[][] S = {
    { 0, 0, 0, 0, 0 },
    { 0, 0, 0, 0, 0 },
    { 0, 0, 0, 0, 0 },
    { 0, 0, 0, 0, 0 },
    { 0, 0, 0, 0, 0 }
};
```

After calling fillGrid(LL, UR, S), S should contain

```
{
  { 0, 11, 12, 13, 14 },
  { 1,  0, 15, 16, 17 },
  { 2,  3,  0, 18, 19 },
  { 4,  5,  6,  0, 20 },
  { 7,  8,  9, 10,  0 }
}
```

(The last two elements of LL are excess and therefore ignored.)

```
1  /** Fill the lower-left triangle of S with elements of LL and the
2   * upper-right triangle of S with elements of UR (from left-to
3   * right, top-to-bottom in each case). Assumes that S is square and
4   * LL and UR have at least sufficient elements. */
5  public static void fillGrid(int[] LL, int[] UR, int[][] S) {
6      int N = S.length;
7      int kL, kR;
8      kL = kR = 0;
9
10     for (int i = 0; i < N; i += 1) {
11
12         -----
13
14         -----
15
16         -----
17
18         -----
19
20         -----
21
22         -----
23
24         -----
25
26         -----
27
28         -----
29     }
30 }
```