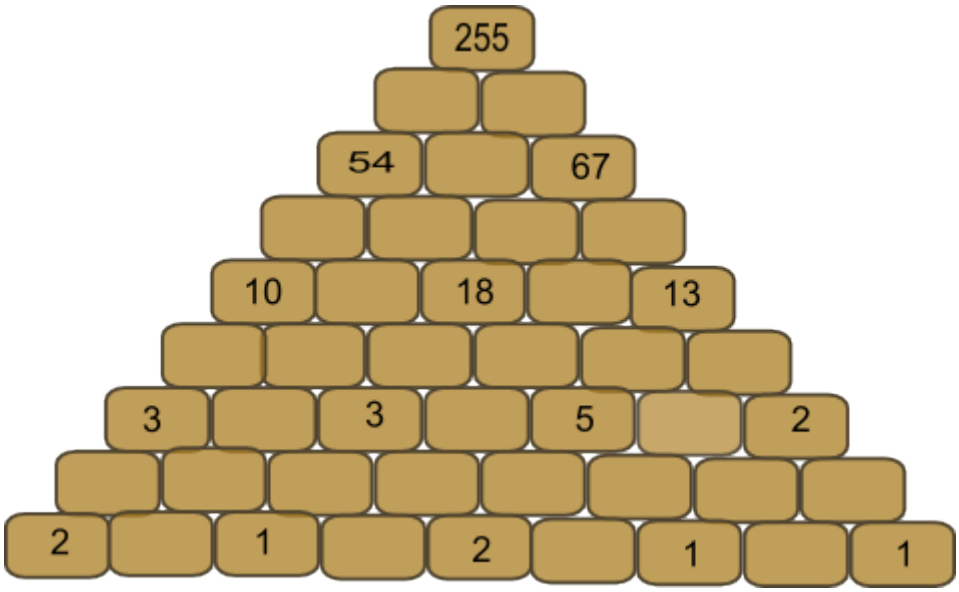
UVa Online Judge

11040 Add bricks in the wall

This in not "*another brick in the wall*", it's just a matter of adding numbers. Suppose you have a wall with the shape of a triangle, like the one shown below. The wall has 9 rows and row i has exactly i bricks, considering that top row is the first one and bottom row is the ninth. Some bricks are labeled with a number and other ones are blank. Notice that labeled bricks appear only on odd rows and they occupy odd positions within the row. The problem you must solve is finding a suitable number for each blank brick taking into account one simple rule: the number of a brick is obtained by adding the numbers of the two bricks below it. Obviously, this rule does not apply to the ninth row. Numbers are supposed to be integers.



Input

The first line of the input contains an integer N, indicating the number of test cases. This line is followed by the lines corresponding to the test cases. Each test case is described in five lines. These five lines correspond to odd rows of the wall, from top to bottom, as described above. Line i contains the numbers corresponding to odd bricks on row i of the wall (that is, non blank bricks), enumerated from left to right and separated with a single space. It is supposed that each test case is correct, that is, there exists a solution to the problem that the case describes.

Output

For each test case, the output should consist of nine lines describing the numbers of all bricks of the wall. So, line i should contain the numbers corresponding to the i bricks on row i of the wall, enumerated from left to right and separated by a single space.

Note: Here we have an example with two test cases. The first one corresponds to the wall depicted above.

Sample Input

```
2
255
54 67
10 18 13
3 3 5 2
2 1 2 1 1
256
64 64
16 16 16
4 4 4 4
1 1 1 1 1
```

Sample Output

```
255
121 134
54 67 67
23 31 36 31
10 13 18 18 13
5 5 8 10 8 5
3 2 3 5 5 3 2
2 1 1 2 3 2 1 1
2 0 1 0 2 1 1 0 1
256
128 128
64 64 64
32 32 32 32
16 16 16 16 16
8 8 8 8 8 8
4 4 4 4 4 4 4
2 2 2 2 2 2 2 2
1 1 1 1 1 1 1 1 1
```