

[account](#) | [logout](#)[Welcome](#) [Rubric](#) [Competitions](#)[Hide menu](#)[2023-2024 ACSL 第四轮正式考试 \(Junior\)](#)

ACSL 2248

Time Remaining 1:59 hrs:mins

[ACSL 2248](#)

CLASS / SOURCE NAME

Nameyourclass acsljr.

[End competition](#)

PROBLEM: The ACSL version of the popular 2248 numbers game uses an 8 x 5 board with tiles containing powers of 2. Initially, the board contains a random set of tiles with values up to and including 256. You'll be given a path that exists on the board that will be given as a list of locations of the tiles on the path. Each location is a 2-digit number representing the row and column. For example, the number 24 is row 2, column 4 which has a value of 256 on the board shown here.

| | | | | |
|-----|-----|-----|-----|----|
| 4 | 128 | 4 | 128 | 32 |
| 16 | 16 | 4 | 256 | 16 |
| 32 | 4 | 16 | 64 | 4 |
| 8 | 64 | 64 | 256 | 8 |
| 16 | 2 | 2 | 256 | 4 |
| 32 | 128 | 2 | 64 | 8 |
| 256 | 32 | 128 | 16 | 2 |
| 8 | 32 | 32 | 4 | 32 |

Your program will modify the board as follows:

1. The last tile in the path is replaced by the power of 2 that is the smallest power of 2 that is greater than or equal to the sum of all of the tiles in the path.
2. The other tiles on the path are removed from the board.
3. The tiles above the removed tiles drop down and are replaced by new tiles.
4. New tiles are added to the board in row-major order, starting at the top left, using decreasing powers of 2, from 256 down to 2, cycling back to 256 if needed.

Output the modified board.

EXAMPLE:

In the board shown above, the path you are given is 13 23 32 41 51 61 72 82 83. This is the path 4-4-4-8-16-32-32-32-32, as shown in the following left image, which has a sum of 164. The next higher power of 2 is 256. Remove all of the tiles from the grid and replace the last 32 with a 256 and then shift all columns down as shown in the middle image. Replace the empty locations in row-major order with the powers of 2 from 256 down to 2, circulating if needed as shown in the right image below.



This is the final board that should be output, in row-major order:

256 128 64 128 32 32 16 8 256 16 4 2 16 64 4 4 128 64 256 8 16 16 2 256 4 32 64 2 64 8 256 2 128 16 2 8 128 256 4 32

INPUT: There are two strings: the first one represents the board of 40 numbers given in row-major order, and the second one is the path that exists on that board. All numbers in the first string will be powers of 2 between 2 and 256, inclusive. All numbers in the second string will be 2-digit numbers representing the location of each tile on the path, given as the row (1-8) and column (1-5). Every number in each string is separated by a single space.

OUTPUT: Output the final board as a string of 40 numbers in row-major order, each separated by a single space.

| SAMPLE INPUT | SAMPLE OUTPUT |
|--|--|
| 4 128 4 128 32 16 16 4 256 16 32 4 16 64 4 8 64 64 256 8 16 2 2 256 4 32 128 2 64 8 256 32 128 16 2 8 32 32 4 32 13 23 32 41 51 61 72 82 83 | 256 128 64 128 32 32 16 8 256 16 4 2 16 64 4 4 128 64 256 8 16 16 2 256 4 32 64 2 64 8 256 2 128 16 2 8 128 256 4 32 |
| 256 128 64 128 32 32 16 8 256 16 4 2 16 64 4 4 128 64 256 8 16 16 2 256 4 32 64 2 64 8 256 2 128 16 2 8 128 256 4 32 51 52 61 62 73 82 83 | 256 128 64 128 32 32 16 64 256 16 256 8 8 64 4 32 128 16 256 8 4 16 64 256 4 4 2 2 64 8 256 128 2 16 2 8 2 1024 4 32 |
| 256 16 256 2 32 2 16 2 16 8 32 2 256 64 16 4 2 128 2 32 8 8 32 256 2 2 4 8 32 128 16 16 32 64 256 4 16 128 4 8 14 23 32 42 41 51 52 63 72 73 74 83 | 256 128 64 32 32 16 8 4 2 8 256 256 128 16 16 2 64 256 64 32 32 16 256 2 2 2 16 128 256 128 16 4 32 32 256 4 16 512 4 8 |
| 256 16 256 2 32 2 16 2 16 8 32 2 256 64 16 4 2 128 2 32 8 8 32 256 2 8 4 8 32 128 16 16 32 64 256 4 16 128 4 8 14 23 32 41 51 52 61 71 72 82 | 256 128 64 32 32 16 8 256 16 8 4 2 256 64 16 256 16 128 2 32 256 16 32 256 2 2 2 8 32 128 32 4 32 64 256 4 128 128 4 8 |

256 8 4 16 128 64 4 32 256 256 32 32 8 4 2 64 128 8 2 8 64
32 64 16 128 4 4 4 16 64 64 2 8 8 32 128 128 128 64 4
13 22 33 43 54 64 75 84 83 82 81

256 256 128 64 32 64 16 8 4 128 32 8 2 256 256 64 32 128 16
2 64 128 32 256 8 4 32 64 4 128 64 4 4 2 64 1024 2 8 8 4

-----以下为中文翻译，仅供参考-----

问题：知名的2248数字游戏（ACSL版）采用一个 8×5 的棋盘，棋盘中每个方块上的数字都是 2 的幂次方值。游戏开始时，一系列数字方块随机放置在棋盘上，方块上的数字最大可达 256（包括 256）。玩家会被给定一条棋盘上的路径，表示为路径中所有方块的位置。方块位置用一个 2 位数表示，每位数字分别代表方块所在行和列。例如，数字 24 代表棋盘上第 2 行、第 4 列的方块，表示的数字为 256。



玩家需要按照如下要求调整棋盘：

- 1. 路径中最后一个方块用大于或等于路径中所有数字方块之和的 2 的最小次幂代替。

2. 将路径中其他数字方块从棋盘移除。
3. 当某个方块被移除，其上方块落下填补空位，同时方块落下产生的空位被新的方块替代。
4. 从棋盘左上角开始，按行优先顺序添加新的数字方块。新添加方块上的数字是从 256 递减至 2 的 2 的幂次方值。如有需要可循环回到 256 重新开始递减。

输出调整后的棋盘。

示例：

对于上图所示棋盘，玩家被给定的路径是13 23 32 41 51 61 72 82 83，代表棋盘中 4-4-4-8-16-32-32-32-32 数字方块，如左下图所示。这些数字方块的总和为 164。下一个 2 的更高次幂是 256。从棋盘中移除该路径中所有方块，并用 256 方块来替代最后的 32 方块。随后将所有列上的方块下移，如中间图所示。按行优先顺序，用数字为 2 的幂次方值的方块来替代棋盘中的空白位置，其中数字范围为从 256 递减到 2。如有需要可循环回到 256重新开始递减，如右下图所示。



右上图为最终棋盘，按行优先顺序输出：

256 128 64 128 32 32 16 8 256 16 4 2 16 64 4 4 128 64 256 8 16 16 2 256 4 32 64 2 64 8 256 2 128 16 2 8 128 256 4 32

输入：输入两行字符串：第一行字符串包含棋盘上的 40 个数字，这些数字按行优先顺序排列；第二行字符串为棋盘上的给定路径。第一行字符串中的所有数字都是 2 的幂次方，介于 2 到 256 之间（包括 2 和 256）；第二行字符串都是表示路径中每个方块位置的两位数，其中每位数字分别代表行数(1-8)和列数（1-5）。每行字符串中的每个数字之间用一个空格隔开。

输出：按行优先顺序将最终棋盘输出为一行包含 40 个数字的字符串，每个数字之间用一个空格隔开。

| 样例输入 | 样例输出 |
|--|--|
| 4 128 4 128 32 16 16 4 256 16 32 4 16 64 4 8 64 64 256 8 16 2 2 256 4 32 128 2 64 8 256 32 128 16 2 8 32 32 4 32 13 23 32 41 51 61 72 82 83 | 256 128 64 128 32 32 16 8 256 16 4 2 16 64 4 4 128 64 256 8 16 16 2 256 4 32 64 2 64 8 256 2 128 16 2 8 128 256 4 32 |
| 256 128 64 128 32 32 16 8 256 16 4 2 16 64 4 4 128 64 256 8 16 16 2 256 4 32 64 2 64 8 256 2 128 16 2 8 128 256 4 32 51 52 61 62 73 82 83 | 256 128 64 128 32 32 16 64 256 16 256 8 8 64 4 32 128 16 256 8 4 16 64 256 4 4 2 2 64 8 256 128 2 16 2 8 2 1024 4 32 |
| 256 16 256 2 32 2 16 2 16 8 32 2 256 64 16 4 2 128 2 32 8 8 32 256 2 2 4 8 32 128 16 16 32 64 256 4 16 128 4 8 14 23 32 42 41 51 52 63 72 73 74 83 | 256 128 64 32 32 16 8 4 2 8 256 256 128 16 16 2 64 256 64 32 32 16 256 2 2 2 16 128 256 128 16 4 32 32 256 4 16 512 4 8 |
| 256 16 256 2 32 2 16 2 16 8 32 2 256 64 16 4 2 128 2 32 8 8 32 256 2 8 4 8 32 128 16 16 32 64 256 4 16 128 4 8 14 23 32 41 51 52 61 71 72 82 | 256 128 64 32 32 16 8 256 16 8 4 2 256 64 16 256 16 128 2 32 256 16 32 256 2 2 2 8 32 128 32 4 32 64 256 4 128 128 4 8 |
| 256 8 4 16 128 64 4 32 256 256 32 32 8 4 2 64 128 8 2 8 64 32 64 16 128 4 4 4 16 64 64 2 8 8 32 128 128 128 64 4 13 22 33 43 54 64 75 84 83 82 81 | 256 256 128 64 32 64 16 8 4 128 32 8 2 256 256 64 32 128 16 2 64 128 32 256 8 4 32 64 4 128 64 4 4 2 64 1024 2 8 8 4 |

注意：

- (1) 样本数据仅为部分测试数据，测试用例全部通过不代表通过本题。
- (2) 你必须通过数据库中所有的测试点才能获得该题满分。
- (3) java 语言里面的 class name（类名）需要用本题的 Source file name（即：acsljr）。
- (4) 平台判分规则为调取 5 次 input 值，每次单独判分。

Compiler Python 3/CPython ▼

1 Enter your program code here

Enter custom input data here (optional)

Run code

Submit code

Previous submissions

Click the filename to download

Date Filename Score Status Compiler

Asdan Challenges are proud to use [The Cuttle Platform](#) to host our challenges.
The Cuttle system can be used for maths and science competitions, entrance exams or employee exams.