**PROBLEM:** The ACSL version of the popular 2248 numbers game uses an 8 x 5 board with tiles containing the powers of 2. Initially, the board contains a random set of tiles with values up to and including 256.

图片包含 图示

描述已自动生成

You need to find a valid path on the board that obeys the following rules:

   1. The first 2 tiles in the path must have the same value. Look for the starting tiles in row-major order, starting in the upper left. Look for the second tile in the order →, ↙, ↓, ↘.

   2. Each new tile added to the path must have the same value as the previous tile or the next increasing power of 2, whichever comes first, looking at the 5 neighbors in the following order: ←, →, ↙, ↓, ↘.

  3. Continue adding tiles as long as possible.

  4. A tile cannot be used more than once on a path.

After you have found the path, do the following:

  1. All tiles in the path except the last one are removed from the board.

  2. The final 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.

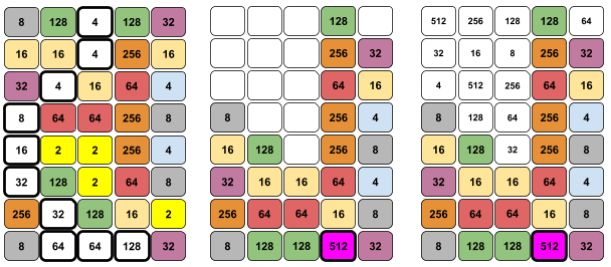
  3. Only 8 powers of 2 can exist on the board at any time. If the new tile added is larger than the largest power of 2 on the board already, then it becomes the largest tile that can be used to replenish the board. For example, if 1024 = 210is added to the board, all 2s and 4s are also removed from the board, since the only eight powers of 2 that can be on the board are 1024, 512, 256, 128, 64, 32, 16, and 8.

  4. All of the tiles in a column above empty locations are moved down to fill those empty locations.

  5. Finally, the empty locations at the top of the grid are filled in row-major order with new tiles with values using 8 decreasing powers of 2, cycling back to the largest number if necessary

*We guarantee that a valid path will be found.*

**EXAMPLE:** In the board shown above, the path you find is 4-4-4-8-16-32-32-64-64-128 which is highlighted in the first image below. The sum of the tiles in the path is 356. In the second image, the final tile in the path has been replaced with 512, the power of 2 that is greater than or equal to 356, and all other tiles in the path have been removed. Because this is 29, all 2s are removed as well. The remaining tiles in each column shift downward. Finally, in the third image, all empty locations have been replaced with new tiles from 512 down to 4.



The final board is output as:

512 256 128 128 64 32 16 8 256 32 4 512 256 64 16 8 128 64 256 4 16 128 32 256 8 32 16 16 64 4 256 64 64 16 8 8 128 128 512 32

**INPUT:** There is one string of 40 numbers, all powers of 2, from 2 to 256 inclusive, each separated by a single space.

**OUTPUT:** Output the board after playing the game once. The board is output as a string of 40 numbers in row-major order, each separated by a single space.

|  |  |
| --- | --- |
| **SAMPLE INPUT** | **SAMPLE OUTPUT** |
| 8 128 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 64 64 128 32 | 512 256 128 128 64 32 16 8 256 32 4 512 256 64 16 8 128 64  256 4 16 128 32 256 8 32 16 16 64 4 256 64 64 16 8 8 128  128 512 32 |
| 256 128 64 128 32 32 16 8 256 16 4 2 16 64 4 4 128 32 256 8  16 16 64 256 4 32 64 2 64 8 256 2 128 16 2 8 128 256 4 32 | 1024 512 256 128 64 32 16 8 128 1024 256 512 256 256 128 32  64 32 64 32 16 16 8 256 16 32 128 64 256 8 256 128 8 64 8 8  16 1024 16 32 |
| 256 16 256 2 32 2 32 2 16 8 32 2 256 64 16 4 2 128 2 32 8 8  32 256 2 2 4 8 32 128 2 16 32 64 256 4 2 128 4 8 | 512 256 128 64 32 16 8 4 512 32 256 128 64 32 8 16 8 256 16  16 4 512 256 64 32 256 16 128 256 128 32 32 32 32 256 4 4  512 4 8 |
| 8 8 16 64 64 256 2 128 16 4 4 64 4 64 16 256 16 64 64 32 32  64 64 256 128 8 128 64 2 16 4 16 256 4 8 64 256 32 16 64 | 1024 512 256 128 64 32 16 8 1024 512 256 128 64 32 16 256 8  1024 512 64 256 64 256 64 128 32 16 128 64 16 8 16 64 256 8  64 1024 32 16 64 |
| 4 16 8 2 32 2 2 8 32 4 2 16 16 4 128 128 32 4 2 128 128 64  8 128 128 4 2 16 32 16 8 8 128 64 32 32 8 128 2 128 | 512 256 128 64 32 32 16 8 4 4 4 512 256 128 128 128 64 4 32  128 512 32 8 4 128 4 16 16 128 16 8 8 128 32 32 32 8 128 64  128 |

--------------------------------------------------------------------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 x 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，方块上数字最大可达 256 （包括 256）。

图片包含 图示

描述已自动生成

玩家需要按照以下规则在棋盘上找到一条有效路径：

  1. 该路径中前两个方块上的数字必须相同。从棋盘左上角开始，按行优先顺序，寻找路径的第一个方块。然后按照 →, ↙, ↓, ↘ 顺序寻找第二个方块。

  2. 路径中每个新增方块上的数字必须与前一个方块上的数字相同或者等于 2 的下一个更高次幂。不论哪个方块先出现，按照 ←, →, ↙, ↓, ↘ 顺序观察 5 个相邻数字方块，选用 能达到更好效果的数字方块。

  3. 持续添加方块，尽可能使路径更长。

  4. 一条路径中一个方块只能使用一次。

找到有效路径后，按以下步骤操作：

  1. 将路径中所有方块从棋盘中移除，保留最后一个方块。

  2. 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。

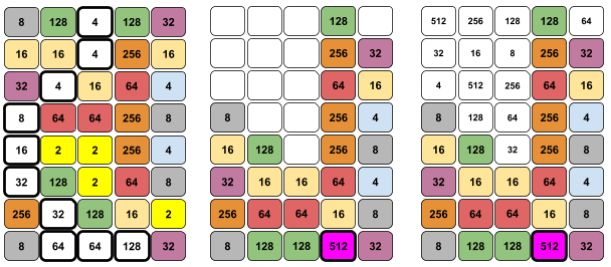
  3. 棋盘中方块上的数字任何时候都只能存在 8 种不同的 2 的幂次方值。如果新增方块上的数字大于棋盘上已经存在的 2 的最大次幂，那么这个用来填补棋盘空白的新增方块成为最大值方块。例如，如果在棋盘中加入1024 = 210方块，那么所有数字为 2 和 4 的方块都将被移出棋盘，因为棋盘中方块上的数字任何时候都只能存在 8 种不同的 2 的幂次方值，即1024、512、256、128、64、32、16、8。

  4. 每一列空白位置上方的所有方块下移，以填补空白。

  5. 最后，按行优先顺序，用新方块填补棋盘顶行中的空白位置。新方块上的数字为 8 个递减的 2 的幂次方的值。如有需要可循环返回最大值重新开始递减。

*我们保证棋盘中一定可以找到有效路径。*

**示例：**对于上述棋盘，玩家找到的路径为 4-4-4-8-16-32-32-64-64-128 ， 如左下图所示。该路径中所有方块之和为 356。在中间图中，用大于或等于 356 的 2 的幂次方，即 512，来替代路径中的最后一个方块，并移除路径中其他方块。因为 512 = 29，且棋盘中方块上的数字任何时候都只能存在 8 种不同的 2 的幂次方值，所以也要移除所有数字为 2 的方块。每一列中的剩余方块下移。最后，在右下图中，用新数字方块填补所有的空白位置，数字为从 512 递减到 4 的 2 的幂次方的值。



最终棋盘输出为：

512 256 128 128 64 32 16 8 256 32 4 512 256 64 16 8 128 64 256 4 16 128 32 256 8 32 16 16 64 4 256 64 64 16 8 8 128 128 512 32

**输入：**输入一行包含 40 个数字的字符串，所有数字都是介于 2 到 256（包括 2 和 256）之间的 2 的幂次方。每个数字之间用一个空格隔开。

**输出：**一轮游戏结束后按行优先顺序，将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** |
| 8 128 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 64 64 128 32 | 512 256 128 128 64 32 16 8 256 32 4 512 256 64 16 8 128 64  256 4 16 128 32 256 8 32 16 16 64 4 256 64 64 16 8 8 128  128 512 32 |
| 256 128 64 128 32 32 16 8 256 16 4 2 16 64 4 4 128 32 256 8  16 16 64 256 4 32 64 2 64 8 256 2 128 16 2 8 128 256 4 32 | 1024 512 256 128 64 32 16 8 128 1024 256 512 256 256 128 32  64 32 64 32 16 16 8 256 16 32 128 64 256 8 256 128 8 64 8 8  16 1024 16 32 |
| 256 16 256 2 32 2 32 2 16 8 32 2 256 64 16 4 2 128 2 32 8 8  32 256 2 2 4 8 32 128 2 16 32 64 256 4 2 128 4 8 | 512 256 128 64 32 16 8 4 512 32 256 128 64 32 8 16 8 256 16  16 4 512 256 64 32 256 16 128 256 128 32 32 32 32 256 4 4  512 4 8 |
| 8 8 16 64 64 256 2 128 16 4 4 64 4 64 16 256 16 64 64 32 32  64 64 256 128 8 128 64 2 16 4 16 256 4 8 64 256 32 16 64 | 1024 512 256 128 64 32 16 8 1024 512 256 128 64 32 16 256 8  1024 512 64 256 64 256 64 128 32 16 128 64 16 8 16 64 256 8  64 1024 32 16 64 |
| 4 16 8 2 32 2 2 8 32 4 2 16 16 4 128 128 32 4 2 128 128 64  8 128 128 4 2 16 32 16 8 8 128 64 32 32 8 128 2 128 | 512 256 128 64 32 32 16 8 4 4 4 512 256 128 128 128 64 4 32  128 512 32 8 4 128 4 16 16 128 16 8 8 128 32 32 32 8 128 64  128 |

窗体顶端