nanolabs
Prototyping the future

Statement: You have probably seen screens that look like this:

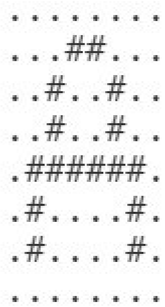
How do these panels work? Pretty simple. You might have heard of matrices before. These panels simply act like a matrix, each pixel (pixel = a single LED) representing a [i, j] position which can be either enabled or disabled in order to display the desired characters. To reduce the memory used and the processing power needed, a byte might contain the state of 8 pixels in a row, which basically means that if we have an 8x8 panel containing 64 pixels, we only need 8 bytes to store all the necessary states. As an Embedded Software Developer, you might have to work with this kind of technology.

Problem: You are given a theoretical 8x8 LED panel and an array of 8 bytes, each byte representing a row of the panel. Your task is to take the array of bytes and output it as a matrix of characters, where 1 is represented by ‘ # ’ and 0 is represented by ‘ . ’. The preferred way of solving this problem is by using bit shifting and bitwise operations.

Input:
Example 1: int A[] = {0x00, 0x18, 0x24, 0x24, 0x7E, 0x42, 0x42, 0x00};
Example 2: int B[] = {0x00, 0x3f, 0x41, 0x61, 0x3f, 0x61, 0x41, 0x3f};

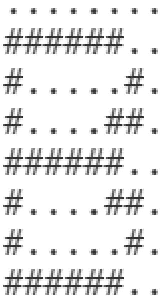
Output:

Output for example 1:



```
........
...##...
..#..#..
..#..#..
.######.
.#....#.
.#....#.
........
```

Output for example 2:



```
........
######..
#.....#.
#....##.
######..
#....##.
#.....#.
######..
```

Note:

- You may use any resources to solve this problem.
- Please include any assumptions you have made in the code as comments.