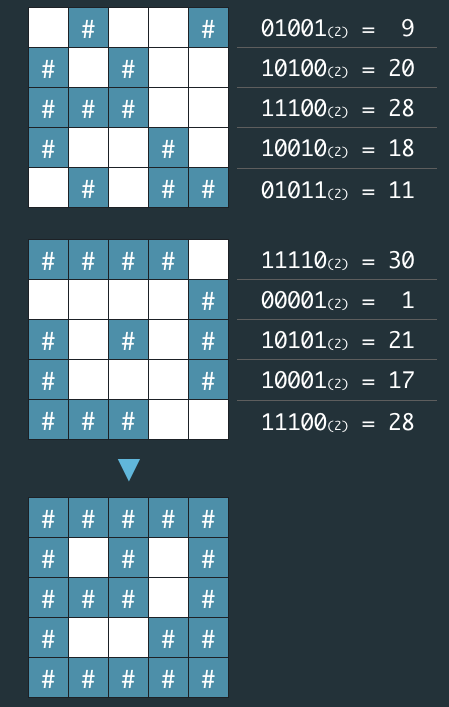
문제 설명

**비밀지도**

네오는 평소 프로도가 비상금을 숨겨놓는 장소를 알려줄 비밀지도를 손에 넣었다. 그런데 이 비밀지도는 숫자로 암호화되어 있어 위치를 확인하기 위해서는 암호를 해독해야 한다. 다행히 지도 암호를 해독할 방법을 적어놓은 메모도 함께 발견했다.

1. 지도는 한 변의 길이가 n인 정사각형 배열 형태로, 각 칸은 "공백"(" ") 또는 "벽"("#") 두 종류로 이루어져 있다.
2. 전체 지도는 두 장의 지도를 겹쳐서 얻을 수 있다. 각각 "지도 1"과 "지도 2"라고 하자. 지도 1 또는 지도 2 중 어느 하나라도 벽인 부분은 전체 지도에서도 벽이다. 지도 1과 지도 2에서 모두 공백인 부분은 전체 지도에서도 공백이다.
3. "지도 1"과 "지도 2"는 각각 정수 배열로 암호화되어 있다.
4. 암호화된 배열은 지도의 각 가로줄에서 벽 부분을 1, 공백 부분을 0으로 부호화했을 때 얻어지는 이진수에 해당하는 값의 배열이다.



네오가 프로도의 비상금을 손에 넣을 수 있도록, 비밀지도의 암호를 해독하는 작업을 도와줄 프로그램을 작성하라.

**입력 형식**

입력으로 지도의 한 변 크기 n 과 2개의 정수 배열 arr1, arr2가 들어온다.

* 1 ≦ n ≦ 16
* arr1, arr2는 길이 n인 정수 배열로 주어진다.
* 정수 배열의 각 원소 x를 이진수로 변환했을 때의 길이는 n 이하이다. 즉, 0 ≦ x ≦ 2n - 1을 만족한다.

**출력 형식**

원래의 비밀지도를 해독하여 '#', 공백으로 구성된 문자열 배열로 출력하라.

**입출력 예제**

| **매개변수** | **값** |
| --- | --- |
| n | 5 |
| arr1 | [9, 20, 28, 18, 11] |
| arr2 | [30, 1, 21, 17, 28] |
| 출력 | ["#####","# # #", "### #", "# ##", "#####"] |

| **매개변수** | **값** |
| --- | --- |
| n | 6 |
| arr1 | [46, 33, 33 ,22, 31, 50] |
| arr2 | [27 ,56, 19, 14, 14, 10] |
| 출력 | ["######", "### #", "## ##", " #### ", " #####", "### # "] |

// Cacao\_Code\_Test.cpp : 콘솔 응용 프로그램에 대한 진입점을 정의합니다.

//

#include "stdafx.h"

#include <iostream>

#include <string>

#include <vector>

#include <stack>

using namespace std;

stack<int> binary(int num, int n)

{

stack<int> binStack;

for (int i = 0; i < n; i++)

{

if (num > 0)

binStack.push(num % 2);

else

binStack.push(0);

num /= 2;

}

return binStack;

}

string compare(stack<int> bin1, stack<int> bin2)

{

string tmp;

int nSize = bin1.size();

int a = 0;

int b = 0;

for (int i = 0; i < nSize; i++)

{

a = bin1.top();

b = bin2.top();

if (a || b) tmp += "#";

else tmp += " ";

bin1.pop();

bin2.pop();

}

return tmp;

}

vector<string> solution(int n, vector<int> arr1, vector<int> arr2)

{

stack<int> bin1;

stack<int> bin2;

vector<string> answer;

int nSize = arr1.size();

for (int i = 0; i < nSize; i++)

{

bin1 = binary(arr1[i], n);

bin2 = binary(arr2[i], n);

answer.push\_back(compare(bin1, bin2));

}

return answer;

}

int main()

{

int n = 5;

vector<int> arr1 = {9, 20, 28, 18, 11};

vector<int> arr2 = {30, 1, 21, 17, 28};

solution(n, arr1, arr2);

return 0;

}