Let's define a function f(x,y) of two positive integers x and y as follows:

* First, convert both x and y to binary, and reverse their binary representations;
* Sum up the obtained values;
* Reverse the binary representation of the result.

Given an array of numbers, your task is to construct the *f-plication* table of its values. The result should thus be a square matrix of size numbers.length × numbers.length, with the value at (i, j) equal to f(numbers[i], numbers[j]).

**Example**

For numbers = [42, 61], the output should be

sumBitReverse(numbers) = [[21, 17],

[17, 61]]

Here's why:

* 4210 = 1010102 --reversed--> 101012 = 2110;
* 6110 = 1111012 --reversed--> 1011112 = 4710;
* the value at (0, 0) should be equal to 21 + 21 = 4210 = 1010102 --> 101012 = 2110;
* the value at (1, 1) should be equal to 47 + 47 = 9410 = 10111102 --> 1111012 = 6110;
* the value at (0, 1) should be equal to the value at (1, 0), and calculated as 21 + 47 = 6810 = 10001002 --> 100012 = 1710.

**Input/Output**

* **[time limit] 3000ms (cs)**
* **[input] array.integer64 numbers**

An array of positive integers.  
1 ≤ input.Length ≤ 250,  
1 ≤ input[i] ≤ 251-1.

* **[output] array.array.integer64**

<https://codefights.com/challenge/j336JGvw3qnyBsqpp/solutions>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static long[][] sumBitReverse(long[] numbers)

{

List<string> inversos = new List<string>();

for (int i = 0; i < numbers.Length; i++)

{

char[] bin = Convert.ToString(numbers[i], 2).ToCharArray();

Array.Reverse(bin);

inversos.Add(new string (bin));

// Console.WriteLine(bin);

}

long[][] res = new long[numbers.Length][];

for (int i = 0; i < numbers.Length; i++)

{

res[i] = new long[numbers.Length];

for (int j = 0; j < numbers.Length; j++)

{

long a = Convert.ToInt64(inversos[i], 2);

long b = Convert.ToInt64(inversos[j], 2);

// Console.Write(a + b);

long sum\_ab = a + b;

char[] bin = Convert.ToString(sum\_ab, 2).ToCharArray();

Array.Reverse(bin);

long x = Convert.ToInt64(new string( bin), 2);

// Console.Write(x + " ");

res[i][j] = x;

}

// Console.WriteLine();

}

return res;

}

static void Main(string[] args)

{

long[] numbers = { 42, 61 };

long[][] res = sumBitReverse(numbers);

for (int i = 0; i < res.Length; i++)

{

for (int j = 0; j < res[i].Length; j++)

{

Console.Write(res[i][j] + " ");

}

Console.WriteLine();

}

Console.ReadLine();

}

}

}