# Extreme Encoding Problem Code: **KOL16B**

Lajuk is a little girl who loves playing with array. In her 10th birthday, she got two arrays as presents. Let's call them **A** and **B**. Both arrays have the same size **n** and contains integers between **0** to **30000**.

Lajuk's hard-drive is almost full of presents and she barely has any space to keep the arrays. She discovered a brilliant function to merge the array into one:

```
int encodeInteger(int x, int n){

        n = n<<(1<<(1<<(1<<1)));

        x = x | n;

        return x;

}

void encodeArray(int *A, int *B, int n){

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

            A[i] = encodeInteger(A[i], B[i]);

        }

}
```

Lajuk passed **A** and **B** into the encodeArray function. After that she discarded array **B**and saved the modified version of array **A** in the hard-drive.

Now in her 15th birthday Lajuk wants to play with those arrays again. She found the modified version of array A in the hard-drive but she forgot how to recover the original arrays from it. Being upset, she asked for your help. Can you help her to recover the original arrays?

---

### Input

The first line contains **T (1 ≤ T ≤ 100)**, the number of test cases. The first line of each test cases contains **n (1 ≤ n ≤ $10^4$)**, the size of the array. Next **n** line contains **n** integers denoting the modified array **A**.

## Output

For each case print the case number in the first line. In the second line, print n integers denoting the original array **A**. In the third line print n integers denoting the array **B**. Two consecutive integers should be separated by a single space.

## Sample

**Input**

```
1

4

196613

655370

196620

131079
```

**Output**

```
Case 1:

5 10 12 7

3 10 3 2
```