3493 01000001

Adding binary numbers is a very simple task, and very similar to the longhand addition of decimal numbers. As with decimal numbers, you start by adding the bits (digits) one column at a time, from right to left. Unlike decimal addition, there is little to memorize in the way of rules for the addition of binary bits:

```
0 + 0 = 0
1 + 0 = 1
0 + 1 = 1
1 + 1 = 10
1 + 1 + 1 = 11
```

Just as with decimal addition, when the sum in one column is a two-bit (two-digit) number, the least significant figure is written as part of the total sum and the most significant figure is "carried" to the next left column. Consider the following examples:

```
                    11  1 <-- Carry bits --> 1   11
  1001101             1001001                 1000111
+ 0010010           + 0011001               + 1010110
 --------           ---------               ---------
  1011111             1100010                 10011101
```

The addition problem on the left did not require any bits to be carried, since the sum of bits in each column was either `1` or `0`, not `10` or `11`. In the other two problems, there definitely were bits to be carried, but the process of addition is still quite simple.

Input

The first line of input contains an integer N, $(1 \leq N \leq 1000)$, which is the number of binary addition problems that follow. Each problem appears on a single line containing two binary values separated by a single space character. The maximum length of each binary value is 80 bits (binary digits). Note: The maximum length result could be 81 bits (binary digits).

Output

For each binary addition problem, print the problem number, a space, and the binary result of the addition. Extra leading zeroes must be omitted.

Sample Input

```
3
1001101 10010
1001001 11001
1000111 1010110
```

Sample Output

```
1 1011111
2 1100010
3 10011101
```