# The Brain-Folk Computer

The inhabitants of Brainlandia love rabbits. Every household keeps track of the number of rabbits it owns with great diligence. There are, in fact, exactly 30,000 houses in Brainlandia, all lying in a perfectly straight line on the northern bank of the Tarpit River. The famous Brainlandian magician Corrado likes to go from door to door making rabbits appear out of his hat, thus increasing the number of rabbits in that household. He is only able to make rabbits appear, not disappear, and he can only travel east, because when he tries to travel west the wind catches his hat the wrong way and it loses its magical power.

On the other side of the river lies the strange town of Folklandia. Unlike their neighbors to the north, Folklandians hate rabbits. Employees of the rabbit pest control company Urban Exterminators go from door to door trying to eliminate as many rabbits as they can. They can only travel west because of faulty compasses installed in their vehicles. Naturally they tend to keep away from Brainlandians, for fear of rabbit infestations.

Unfortunately, neither the Brainlandians nor the Folklandians are very good at mathematics. In particular, they have quite a hard time determining how many rabbits they'll have after a visit from Corrado or from Urban Exterminators, as the case may be. If only they had a computer that could tell them the answer...

A traveling key manufacturer named Louis has recently visited the two towns and got to thinking that, with a little cooperation, the towns themselves could act as powerful computers! First of all, Corrado could travel inside the vehicles driven by Urban Exterminators, allowing him to escape the wind and travel west. Additionally, with his magical powers he could fix the compasses and allow the vehicles to travel east. Finally, Louis has invented a very concise programming language to instruct the magician-exterminator team where to go and when to increase or decrease the number of rabbits.

The inhabitants of the two towns are skeptical that Louis's proposal would allow them to compute anything useful. It has been decided that the true test will be whether Louis can provide them with a program to help them keep track of their rabbits. They are more concerned with addition than subtraction, since they are familiar with two's complement. Also, they prefer doing everything in binary because that way they only have to remember two symbols.

Louis has just had a tall glass of freshly squeezed orange juice, and he now feels ready to write the program and prove that his language can do everything he says it can. As all programs must be written on punch cards and those cards are very expensive to make, Louis would like to write as short a program as possible.

**Note:** You can use any programming language you want, as long as it is brainf**k.

## Input

The first line contains an integer **T** ($1 \leq \mathbf{T} \leq 1000$). Then follow **T** lines, each containing integers **A** and **B** ($0 \leq \mathbf{A},\mathbf{B} \leq 2^{100}$) in binary, separated by a single space. Each line, including the last, is terminated by a single newline (linefeed) character, which has ASCII value 10.

## Output

**T** lines containing the value of **A**+**B** in binary.

## Score

The score is the number of brainf**k commands in your program. All other characters are ignored.

## Example

### Input:

```
5
0 0
0 101010
101010 0
101010 101010
1011011101111011111 1011011011101
```

### Output:

```
0
101010
101010
1010100
1011101001010111100
```

## Additional Info

There are two randomly generated data sets, one with **T**=1000 and the other with **T**=500. **A** and **B** are generated independently, and the average number of bits in either is about 50.

My solution at the time of publication has 232 bytes and runs in 0.25s with 1.9M memory footprint. I also have a 230-byte solution for bff 1.0.5 that would run in about 7.7s with a somewhat larger memory footprint.

I have made the C++ source for my test case judge available (probably of interest only to problem setters). It is a minor variation on the standard "Score is source length" judge.

Update: Reduced source to 201 in bff 1.0.3.1 and 191 in bff 1.0.5 as of 2014-11-07.