

One Good Base Deserves Another

Basil and Blaise are very interested in manufacturing sodium hydroxide, ammonia, and other bases. They currently work out of an old basement but are itching to establish a new home base at the base of a tall mountain overlooking the bay. They're meeting with their real estate agent, Bane, to look for a building large enough to house their basic operations. Bane seems like a trustworthy fellow, based on his professional manner and charming smile, but he secretly harbors base intentions. He uses a special bank, Hexcorp, that conducts all of its business in hexadecimal. He is extra careful to make sure all the figures in his contracts use only the digits 0 through 9, even though they are in hexadecimal, in the hopes that his clients will unwittingly agree to his inflated prices so he can keep a hefty share for himself. He has included a notice about his unusual choice of numeric base in the very fine print of his contracts, to protect himself legally and cover all bases.

Fortunately, Basil and Blaise always read contracts very carefully before signing them, and the strange notice catches their attention. But they don't know anything about converting numbers between bases, since before turning to chemistry Basil was a professional baseball player and Blaise was an aspiring bass guitarist, and neither has had much mathematical training. Please help them avoid getting scammed by sending them a program that will allow them to take an integer in one base and convert it into another base. But be careful: Bane has a deep network of spies, and if they intercept your correspondence, Bane may take drastic action. Luckily, his spies don't understand brainf**k, so you can safely send them anything in that language.

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

Input

For clarity, all integers in this section are given in decimal.

The first line contains an integer **T** ($1 \leq T \leq 1000$) expressed in decimal. Then follow **T** lines, each containing 3 space-separated integers: **B1** and **B2** ($2 \leq B1, B2 \leq 35$) expressed in base **36**, followed by **N** ($0 \leq N \leq 35^{35}$) expressed in base **B1**. For bases greater than 10, only uppercase letters are used.

Each line, including the last, is terminated by a single newline (linefeed) character, which has ASCII value 10.

Output

T lines containing **N** expressed in base **B2**. For bases greater than 10, only uppercase letters are allowed.

Example

Input:

```
9
F A 50000
A 2 42
```

2 A 101010
2 Z 1011011101111011111
7 8 0
Z 7 YWX123ABC
Y I ABCDEFG
I Y ABCDEFG
Y Y ABCDEFG

Output:

253125
101010
42
8QQF
0
22400453332065605
1816CB9F4
7X2J66
ABCDEFG

Additional Info

There are two randomly generated data sets, one with $T=1000$ and the other with $T=500$. **B1** and **B2** are generated independently, and the average value of either is about 18.5. The average number of digits in **N** when expressed in decimal is about 14.

My solution at the time of publication has 678 bytes (not golfed) and runs in 3.71s with 1.8M memory footprint.