Secure password (`password`)

For our brand new website we have developed a new custom way to store our hashed password.

To calculate the hash of a password P we performe the following operations:

- Generate two random strings of random length (may be zero) that will be used as salts, respectively called $salt1$ and $salt2$.
- Generate a random permutation $P2$ of the original password P.

The hash value H of the password P is thus $H = salt1 + P2 + salt2$ (where $+$ is the string concatenation operator).

But now we miss the function that given an hash verify the password!

Write a program that, given a set of T pairs of password P and hash H, returns if each password is verified or not.

Input data

Among the attachments you can find templates for each available language that already correctly implements the management of input and output, use them!

Your program must read the input data from the standard input.

The first line of the input contains an integer **T** representing the number of passwords to verify.

For each of the verification requested, the input contains two lines with the respective password P and hash H to verify.

Output data

Among the attachments you can find templates for each available language that already correctly implements the management of input and output, use them!

Your program must write the output data into the standard output.

The output must contains **T** lines representing the results of each password verification. Each line must contains the number **1** if the password is verified, **0** otherwise.

Scoring

For each of the test cases the program will be tested, the following constraints are met:

- $T \leq 5$ and $|P| \leq |H| \leq 10$ for at least 25% of all the test cases.
- $T \leq 10$ and $|P| \leq |H| \leq 100$ for at least 50% of all the test cases.
- $T \leq 15$ and $|P| \leq |H| \leq 1\,000$ for at least 75% of all the test cases.
- $T \leq 20$ and $|P| \leq |H| \leq 10\,000$ for all the test cases.

Examples

input	output
5 pezu eugzupe iagu guiafcr evup hdvuepx ziql qtdzlni quje edjzouq	1 1 1 0 0

Explanation

In the five examples we have:

- H = `eugzupe`, P = `pezu`, salt1 = `eug`, no salt2 and the password is verified.
- H = `guiafcr`, P = `iagu`, no salt1, salt2 = `fcr` and the password is verified.
- H = `hdvuepx`, P = `evup`, salt1 = `hd`, salt2 = `x` and the password is verified.
- H = `qtdzlni`, P = `ziql` and the password is not verified.
- H = `edjzouq`, P = `quje` and the password is not verified.