Tobby, the small and cute dog, wants to prove to the rest of the world he is a descendent of Tutantobby, the great dog pharaoh. As Tutantobby was mummified, his internal organs were dried and separated in different bottles. To prove that Tobby is a descendent of Tutantobby, the DNA of some of these organs must be extracted and compared against Tobby's DNA. This DNA comparison is not a problem for nowadays science, but extracting the DNA from a 5000 years old mummified organ is a real challenge.

The DNA, as you might probably know, is represented as a sequence of letters '`A`', '`G`', '`C`' and '`T`'. All Tobby can expect is to obtain fragments of Tutantobby's DNA corrupted by the years. Tobby requires to assemble Tuntantobby DNA fragments, to do that Tobby takes two DNA fragments and computes the best way to assemble them as in the following example. Lets say Tobby has the two following Tutantobby DNA segments:

- `GATTACCA`
- `TACAACAG`

Note the following alignment and the resulting string:

```
GATTACCA
   TACAACAG

------------
GATTACXACAG
```

The score of this alignment is 4, since there are 4 characters in this alignment that match. The resulting assembled DNA would then be "`GATTACXACAG`", note that the resulting string might have an '`X`' letter representing that Tobby can't tell what that letter would be. Now Tobby wants a program that given only two sequences, outputs the assembled DNA sequence for the alignment with the maximum score. If there are two alignments that produce the same score output the one where the first string is more to the left with respect to the second. For example if the first string is "`ATTG`" and the second is "`GCCA`", the alignment "`ATTGCCA`" is preferable over "`GCCATTG`" since both have a score of 1.

Input

The input consists of several test cases. Each test case contains two strings S_1 and S_2 that indicate the two DNA fragments extracted from Tutantobby. These two strings will only contain the uppercase letters '`A`', '`G`', '`T`' and '`C`'. After each test case there will be a blank line.

- $1 \leq |S_1|, |S_2| \leq 10^5$

Output

For each test case output two lines. The first one with the best alignment score, and on the second line output the respective assembled sequence as explained in the problem statement. If the best score is 0, print on the second line the string '`No matches`'. There should be a blank line after each test case.

Sample Input

```
GATTACCA
TACAACAG

AAAA
GGGG

ATTG
GCCA
```

Sample Output

```
4
GATTACXACAG

0
No matches

1
ATTGCCA
```