Oh!! Mankind is in trouble again. This time, it's a deadly disease spreading at a rate never seen before. The need of the hour is to set up efficient virus detectors. You are the lead at Central Hospital and you need to find a fast and reliable way to detect the footprints of the virus DNA in that of the patient.

The DNA of the patient as well as of the virus consists of lowercase letters. Since the collected data is raw, there may be some errors. You will need to find all substrings in the patient DNA that either exactly match the virus DNA or have at most one mismatch, i.e., a difference in at most one location.

For example, "aa" and "aa" are matching, "ab" and "aa" are matching, while "abb" and "bab" are not.

Function Description

Complete the *virusIndices* function in the editor below. It should print a list of space-separated integers that represent the starting indices of matching substrings in increasing order, or `No match!`.

virusIndices has the following parameter(s):

- *p*: a string that represents patient DNA
- *v*: a string that represents virus DNA

Input Format

The first line contains an integer t, the number of test cases.

. Each of the next t lines contains two space-separated strings p (the patient DNA) and v (the virus DNA).

Constraints

- $1 \leq t \leq 10$
- $1 \leq |p|, |v| \leq 10^5$
- All characters in p and $v \in ascii[a-z]$.

Output Format

For each test case, output a single line containing a space-delimited list of starting indices (0-indexed) of substrings of p which are matching with v according to the condition mentioned above. The indices have to be in increasing order. If there is no matching substring, output `No Match!`.

Sample Input 0

```
3
abbab ba
hello world
banana nan
```

Sample Output 0

```
1 2
No Match!
0 2
```

Explanation 0

For the first case, the substrings of p starting at indices 1 and 2 are "bb" and "ba" and they are matching with the string v which is "ba".
For the second case, there are no matching substrings so the output is `No Match!`.
For the third case, the substrings of p starting at indices 0 and 2 are "ban" and "nan" and they are matching with the string v which is "nan".

Sample Input 1

```
3
cgatcg gc
atcgatcga cgg
aardvark ab
```

Sample Output 1

```
1 3
2 6
0 1 5
```

Explanation 1

For the first case, the substrings of p starting at indices 1 and 3 are "ga" and "gc" and they are matching with the string v which is "gc".
For the second case, the substrings of p starting at indices 2 and 6 are "cga" and "cga" and they are matching with the string v which is "cgg".
For the third case, the substrings of p starting at indices 0, 1 and 5 are "aa", "ar" and "ar" and they are matching with the string v which is "ab".