A router has an in-memory collection B of M external domains which cannot be visited by local network user. There is a rule that if a domain is blocked, any its subdomains are blocked too.

For example, if domain `adult.hb` is blocked, the following domains are blocked too:
`images.adult.hb, ringo.adult.hb, video.ringo.adult.hb`

Write a function (in C#):

```
class Solution { public static int[] solution(string[] A, string[] B); }
```

that, given a non-empty array A of N domains, returns a sequence consisting of L integers where each integer represents an index of a domain in input A array that is not blocked.

For example, given:

```
    A[0] = unlock.microvirus.md     B[0] = microvirus.md

    A[1] = visitwar.com             B[1] = visitwar.de

    A[2] = visitwar.de              B[2] = piratebay.co.uk

    A[3] = fruonline.co.uk           B[3] = list.stolen.credit.card.us

    A[4] = australia.open.com

    A[5] = credit.card.us
```

the function should return the array [1, 3, 4, 5], as explained above.

Assume that:

- N and M are integers within the range [1..20,000];

- L is integer within the range [0..20,000];

- each element of array A is a string with length [2.. 256];

- each element of collection B is a string with length [2..256].

Complexity:

- expected worst-case time complexity is O(N);

- expected worst-case space complexity is O(M), beyond input storage (not counting the storage required for input arguments).

Looping over the list B is not an option. If we check 1000 domains in the list of 1000 blocked domains, we would do 1000000 string comparisons. It would be simply too slow.

Elements of input arrays cannot be modified.