

# Powercode

The longer the code, the stronger the code. The power number (**P**) of a code determines the strength of a code.

While computing the power number of a code we should ignore the keywords in the code. A few sample key words would be *int*, *unsigned*, *string*, *etc*. The power of a code can be determined by the number of characters in the code that **do not** belong to any of the keywords.

More formally, for each keyword **K**, consider all the occurrences of **K** in the code, ignore all these matching substrings while computing the power. See the example for clarity.

**Example:**

If you take a code to be

```
int main() { minteger a; return 0; }
```

The above code has **37** characters in total.

If the key words are

```
int
return
lint
integer
```

Then the power of the above code would be **20, spaces need to be counted:**

```
" main() { " --> 10
"m a; "      --> 5
" 0; }"      --> 5
```

The codes can be concatenated to be made more powerful. If we have two codes **C1**, **C2** with power **P1**, **P2**, the power of code **C1 C2** **\*\***(*here \*\** denotes concatenation) will be **P1 + P2**.

While concatenating two codes, a **%** character gets inseted in between the two codes. For example if **C1** was `print 5;` and **C2** was `print 6;`, **C1 \$ C2 =** `print 5;%print 6;`

You are given **N** codes, **K** keywords, you can use each code as many number of times as you want. You are to output the largest power **X**, that can't be attained by concatinating the codes.

**Note:**

All the following operations are possible.

- Two same codes can be concatenated with each other.
- Some codes need not be used in the concatenation.
- Each code can be used any number of times.

**Input Format**

First line contains **N**. In the following **N** lines, the **i<sup>th</sup>** line represents the code **C<sub>i</sub>**. The following line contains **K** the number of keywords. The following K lines one keyword each.

## Output Format

Print a single line containing **X** as described above.

## Constraints

```
1 <= N <= 10
1 <= |C_i| <= 100000
1 <= K <= 100
1 <= each keyword length <= 1000
```

## Note:

- Key words do not contain spaces. Both the code and the key words do not contain % character. They can contain any other ASCII character.
- If the answer is infinite then print **-1**.

## Sample Input: #1

```
5
lint maint lllint
int maint lllintl
lint maint lllint
lint maint lllint
lint maint lllint
2
int
lint
```

## Sample Output: #1

```
29
```

## Sample Input: #2

```
3
ababaac
babab
abcbab
1
bab
```

## Sample Output: #2

```
5
```

## Explanation:

In sample 1, the powers of codes are 6, 7, 6, 6. And hence the maximum unattainable power by combination is 29. In sample 2, the powers are 4, 0, 3, and hence the result is 5.