Problem A: Assessing Genomes

The world is at the brink of extinction. A mutated virus threatens to destroy all living organisms. As a last hope, a team of super-smart scientists, including – of course – you, is currently working on an antivirus. Unfortunately, your team is unable to analyse the DNA in time. They sequenced n parts of the virus' DNA and need to match them with n available strands for antiviruses. As the algorithms expert, you need to implement a specialised procedure to solve this problem. Your approach needs to be fast – there is not much time left!

You first need to determine the *repetition score* of each DNA sequence. The repetition score of a sequence s is equal to the length of the shortest sequence u such that s is equal to the k-fold repetition of u, for some positive integer k. For instance, `ATGATG` has a repetition score of 3, since it can be produced by repeating `ATG` two times. On the other hand, `ATATA` has a repetition score of 5, as it cannot be produced from any proper substring.

Once you obtained the scores of all sequences, you need to match the n antivirus sequences with the n virus sequences in a way that minimises the damage caused by the virus. When two sequences are matched, the damage caused by the virus is equal to the squared difference between the two repetition scores. For instance, matching the antivirus sequence `ATGATG` with the virus sequence `ATATA` causes $(3-5)^2 = 4$ units of damage.

If you match the DNA sequences optimally, what is the minimal total damage caused by the virus, taken as a sum over all matched pairs?

Input

The input consists of:

- A line with an integer n ($1 \leq n \leq 50$), the number of DNA sequences of the virus and antivirus each.
- n lines, each with a virus DNA sequence.
- n lines, each with an antivirus DNA sequence.

Each DNA sequence is a non-empty string with a length of at most 250 and consists of lowercase letters `a`–`z` and uppercase letters `A`–`Z`.

Output

Output one integer, the minimal total damage.

Sample Input 1

```
2
TTTTTT
TATG
TATATA
AAAGAAAG
```

Sample Output 1

```
1
```

Sample Input 2

```
3
abcdef
aaa
bab
AbAb
xyzxyz
X
```

Sample Output 2

```
10
```

This page is intentionally left (almost) blank.