TASK	LIJEN	BINGO	MRAVOGRAD
input	standard input		
output	standard output		
time limit	1 second	1 second	2 seconds
memory limit	64 MB		
points	**50**	**70**	**80**
	200		

Mirko has recently discovered a telegraph in his attic and now he is transmitting messages to his friend Slavko. The telegraph sends messages using dots and dashes. Mirko has to hold the key for **one second** to send a **dot**, and **two seconds** for a **dash**.

As Mirko is very **lazy** he **changes some symbols** in each word he sends so that he needs the least possible amount of time to transmit a message.

Mirko and Slavko have never heard about Morse code, so they sat down together and made a dictionary of all the words they could ever need. When Mirko sends a message, he makes sure that Slavko can **uniquely** determine the starting word if he goes through the entire dictionary and, of all words of **equal length** as the original word, selects the one which differs from the sent word in the **smallest** number of symbols.

Given the dictionary and the text which Mirko wants to send to Slavko, determine the **smallest total time** required to send all the words (spaces between words are not necessary), so that Slavko can **uniquely decode** the text.

INPUT

The first line contains an integer N ($1 \leq N \leq 2\,000$), the number of words in the dictionary.

The following N lines contains words in the dictionary. Each word is a string of **at most 12** symbols '.' or '-'.

The next line contains an integer L ($1 \leq L \leq 10\,000$), the number of words Mirko sends. The following L lines contain the words Mirko sends, one per line. Each word Mirko sends can be found in the dictionary.

OUTPUT

Output the least time needed for Mirko to send the message, so that Slavko can uniquely decode it.

EXAMPLE TEST CASES

input
```
2
.-..
--.-
2
.-..
--.-
```
output
```
10
```

input
```
3
.-..
-.--
....
2
.-..
-.--
```
output
```
11
```

input
```
3
.-.-.--
.--.--
.-..
3
.-..
.--.--
.-..
```
output
```
14
```

Clarification for first example: Mirko can send **....** instead of the first word, which takes 4 seconds, and instead of the second word he can send **-..-**, which takes another 6 seconds.

In a simplified version of the popular game of Bingo, the host reads numbers and each player looks for those numbers on his card.

Each player has a card which contains all the numbers from 1 to N^2 in N rows and N columns.

The host reads the numbers and the players check if the **last N numbers read** match **one of the rows** on their card. Numbers on the card have to be in the same order as the read numbers. The player gets 1 point for each match.

For example, suppose N is 3 and the player has the following card:

1	**3**	**7**
6	**4**	**5**
2	**8**	**9**

If the host reads the following numbers: 7, 1, 3, 6, 4, 5, 7, 1, 2, 2, 8, 9, 3, then the player gets 2 points, because sequences 6, 4, 5 and 2, 8, 9 appear as rows on his card.

Disappointed with his card for which he got a small number of points, Mirko wonders what is the **largest possible number of points** he could get if the same numbers are read, considering all possible cards.

INPUT

The first line contains two integers N and B ($2 \leq N \leq 4$, $1 \leq B \leq 10\,000$), the size of the card and the number of numbers which the host reads.

The following B lines contain numbers which the host has read. Each of those numbers will be between 1 and N^2.

OUTPUT

Output the largest number of points over all possible cards.

EXAMPLE TEST CASES

input

```
2 11
1
2
1
2
1
2
1
2
3
4
1
```

output

```
5
```

input

```
3 14
1
1
1
1
1
2
3
4
5
6
8
9
9
9
```

output

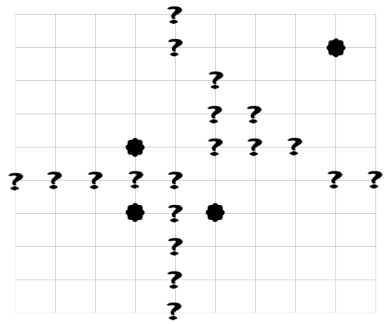
```
2
```

Clarification for first example: one of those cards has 1 2 (contributes 4 points) and 3 4 (1 point) in its rows.

The hard working ants have built a town called Ant Town. They modeled their town after Manhattan, with H horizontal and V vertical streets which cross in V×O intersections. As ants don't like water, with the first raindrops comes chaos in Ant Town. Town authorities have placed umbrellas under which any number of ants can hide, but only on N intersections.

When the rain starts, each ant on an intersection starts running, **using streets**, to the nearest intersection with an umbrella. But, if an ant can choose from more than one such intersection, it panics and, not knowing where to go, **stays on its starting intersection** and gets wet. Town authorities use the name "wet intersections" for such starting intersections.

For example, if Ant Town has 10 horizontal and 10 vertical streets, and if there are 4 intersections with umbrellas, then the question marks in the figure represent "wet intersections":



Picture represents first example. We count streets from left to right from 1 to V and from down upwards from 1 to H.

Write a program which, given the locations of intersections with umbrellas, determines the **number of "wet intersections"** in Ant Town.

INPUT

The first line contains two integers H and V ($1 \leq H, V \leq 30\,000$), the numbers of horizontal and vertical streets in Ant Town.

Horizontal streets are numbered 1 to H, vertical streets 1 to V.

The second line contains an integer N ($1 \leq N \leq 10$), the number of intersections with umbrellas.

Each of the following N lines contains two integer h and v, meaning that there is an umbrella on the crossing of horizontal street h and vertical street v. The locations of all umbrellas will be distinct.

OUTPUT

Output the number of "wet intersections" in Ant Town.

EXAMPLE TEST CASES

input

```
10 10
4
4 4
4 6
6 4
9 9
```

output

```
19
```

input

```
9 9
3
2 2
5 5
8 8
```

output

```
36
```

input

```
100 100
2
50 50
50 51
```

output

```
0
```