TASK	IZBORNIK	KUPUS	PARK
input	standard input		
output	standard output		
time limit (per test case)	1 second		
memory limit (heap+stack)	64 MB		
points	**30**	**50**	**70**
	150		

The menu in a computer program contains N options. Each option is described by one or more words.

Each option in the menu, in order from first to last, is assigned a shortcut – one of the letters in its description. The rules for assigning the shortcut are:

- First consider the **initial letters** of all words in the description, in order from first to last. The first such initial letter not already to another option is selected (no two options may have the same shortcut).
- If all initial letters are already assigned, then consider **all remaining** letters in the description in order, again choosing the first available letter.
- If none of the letters in the description are available, then the option has no shortcut.
- The process is not case sensitive; lowercase and uppercase letters are considered the same.

Write a program which, given the descriptions of all options, determines the shortcuts.

INPUT

The first line contains the integer N ($1 \leq N \leq 30$), the number of options in the menu.

Each of the following N lines contains the description of one option, a sequence of at most 5 words separated by single spaces. Each word will contain at most 10 letters of the English alphabet.

OUTPUT

For each option, in the same order in which they were given, output its description with the shortcut letter surrounded by brackets. If an option is not assigned a shortcut, output its description unchanged. The case of all letters must be the same as in the input.

EXAMPLE TEST DATA

input
```
5
New
Open
Save
Save As
Save All
```

output
```
[N]ew
[O]pen
[S]ave
Save [A]s
Sa[v]e All
```

input
```
8
New window
New file
Copy
Undo
Format
Font
Cut
Paste
```

output
```
[N]ew window
New [f]ile
[C]opy
[U]ndo
F[o]rmat
Fon[t]
Cut
[P]aste
```

Mirko has decided to grow cabbage on the roof of his building. To water the cabbage he will place N identical sprinklers, each watering a circular region with radius 1.

The roof of the building is rectangular, X units long and Y units wide, so it can be modeled by a rectangle in the coordinate plane with sides parallel to the axes and corners in points (0, 0) and (X, Y). Because the pipes are laid out in a perfect grid, all sprinklers are located in points with **integer coordinates**.

Write a program that, given the coordinates of all sprinklers, determines the **total area** of the roof covered by the sprinklers.

INPUT

The first line contains two integers X and Y ($1 \leq X \leq 1000$, $1 \leq Y \leq 1000$), the dimensions of Mirko's roof.

The second line contains the integer N ($1 \leq N \leq 10\,000$), the number of sprinklers.

Each of the following N lines contains the coordinates of one sprinkler; two integers x and y, ($0 \leq x \leq X$, $0 \leq y \leq Y$).

OUTPUT

Output a real number in decimal notation, the overall area of the roof covered by sprinklers, in square units. Your output must be accurate to ± 0.001.

EXAMPLE TEST DATA

input
```
4 5
2
0 0
4 4
```

output
```
2.356194
```

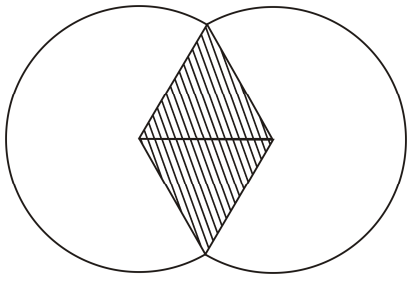
input
```
10 3
2
5 1
6 1
```

output
```
5.054816
```

input
```
7 7
4
3 2
2 2
1 1
6 3
```

output
```
10.767205
```

In the second example, the total area can be calculated as the sum of the areas of two pie slices and two equilateral triangles, giving the expression (the radius of the pie slice as well as the length of the equilateral triangle is 1):

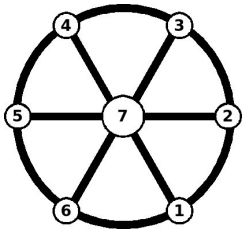


$$P = 2 \cdot \frac{1^2 \cdot \pi}{360^o} \cdot 240^o + 2 \cdot \frac{1^2 \cdot \sqrt{3}}{4} = 2 \cdot \frac{2}{3} \cdot \pi + 2 \cdot \frac{\sqrt{3}}{4}$$

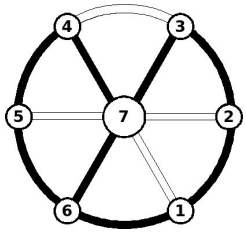
Ivica has decided to take a walk in a nearby park. The park contains a total of N+1 water fountains, the largest one in the center of the park. The remaining fountains are arranged in a circle around the largest one. The surrounding fountains are numbered 1 to N and the central is marked N+1.

The surrounding fountains are connected by paths to form a cycle. Each of the surrounding fountains is also connected to the central fountain, for a total of 2·N paths in the park.

Some of the paths are being cleaned by volunteers and are temporarily unusable.



Example park for N=6. All paths are usable.



Same park with some of the paths unusable.

Ivica starts his walk near some fountain. He proceeds to use paths so that he never visits the same fountain or uses the same path twice. The walk ends when Ivica reaches the fountain where he started.

Write a program that calculates the number of distinct walks Ivica can make. Two walks are different if they don't contain the same paths (so the starting fountain and order of traversal don't matter). For the park in the right image above, there are three walks: 1-2-3-7-6-1, 1-2-3-7-4-5-6-1, and 4-5-6-7-4.

INPUT

The first line contains an integer N ($2 \leq N \leq 100\,000$), the number of fountains other than the central.

Each of the following two lines contains a string of N characters '0' and '1' describing the availabilities of the paths. A zero in some position represents an unavailable path, while a one represents an available path. The two strings are:

1. The outer paths, connecting the surrounding fountains. The paths are given in counterclockwise order, starting with the path connecting fountains N and 1.
2. The inner paths, connecting surrounding fountains to the central one. The paths are again given in counterclockwise order, starting with the path connecting fountain 1 to the central fountain.

OUTPUT

Output the number of distinct walks.

EXAMPLE TEST DATA

input 3 111 111 **output** 7	**input** 6 111011 001101 **output** 3