TASK	TABLICA	ZBROJ	EKIPA	ZNANSTVENIK	DIFERENCIJA	MONO

source code	`tablica.pas`
	`tablica.c`
	`tablica.cpp`
		`zbroj.pas`
		`zbroj.c`
		`zbroj.cpp`
			`ekipa.pas`
			`ekipa.c`
			`ekipa.cpp`
				`znan.pas`
				`znan.c`
				`znan.cpp`
					`dif.pas`
					`dif.c`
					`dif.cpp`
						`mono.pas`
						`mono.c`
						`mono.cpp`

input	standard input (*stdin*)					
output	standard input (*stdin*)					
time limit	1 second	1 second	1 second	1 second	1 second	5 seconds
memory limit	32 MB	32 MB	32 MB	32 MB	64 MB	64 MB
point value	**30**	**50**	**60**	**100**	**120**	**140**
	500					

Perica was always very good at math. His only weak points were addition and division. To help him with that, his teacher presented him with the following problem.

She gave him a 2 by 2 table, containing **positive** integers **A**, **B**, **C** and **D**.

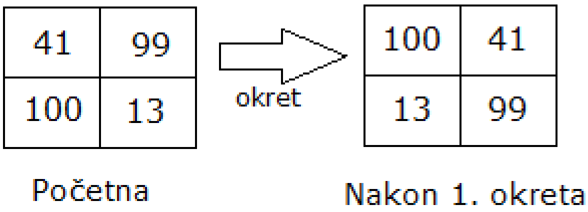
A	**B**
C	**D**

We say that the **value** of a table is equal to:

$$\frac{A}{C} + \frac{B}{D}$$

Perica's task is to find the **minimum** number of 90 degrees clockwise rotations required to maximize the value of a given table.

Result of a single clockwise rotation is shown below.



INPUT

The first line of input contains two space separated integers, **A** and **B**.

The second line of input contains two space separated integers, **C** and **D**.

All integers are **positive** and not greater than 100.

OUTPUT

The first and only line of output must contain a single integer, minimum number of clockwise rotations required to maximize the table's value.

SAMPLE TESTS

input 1 2 3 4 **output** 2	**input** 5 9 7 2 **output** 0	**input** 41 99 100 13 **output** 1

Task ZBROJ

Author: Goran Gašić

After he got tired of rotating tables, the teacher gave Perica another problem. She wrote two integers, **A** and **B**, on the blackboard and asked him to add them.

Perica is never wrong with his calculation, but sometimes he doesn't copy the numbers correctly. The only mistake he ever makes is copying a '5' as a '6', and viceversa.

Given two numbers, **A** and **B**, calculate the minimum and the maximum sum Perica could possibly get.

INPUT

The first and only line of input contains positive integers **A** and **B** (1 ≤ **A**, **B** ≤ 1 000 000).

OUTPUT

In single line of output, print two space separated integers, minimum and maximum sum Perica could get.

SAMPLE TESTS

input	**input**	**input**
11 25	1430 4862	16796 58786
output	**output**	**output**
36 37	6282 6292	74580 85582

In a galaxy far, far away, a programming contest is taking place. Your task is to choose the participants!

N students have applied, and each one of them has some **knowledge** in each of **M** different categories. Knowledge can be represented as a real number. You can send at most **K** students to the contest, but no student can compete in **more than one** category. Multiple students can compete in a single category.

For each student, their knowledge of each category is given.

Choose participants for a contest and categories they will compete in, so that the sum of knowledge is maximal.

INPUT

The first line of input contains integers **N**, **M** and **K** (1 ≤ **M** ≤ 100, 1 ≤ **K** ≤ **N** ≤ 100).

Each of the next **M** lines describes knowledge for one category.

In each line, there are **N** pairs (**i**, **s**), where **i** is the index of the student, and **s** is a positive real number representing their knowledge of corresponding category (0 ≤ **s** ≤ 10). Pairs are given in order of **decreasing** knowledge. Students are numbered from 1 to **N**.

In each line, every student will apear exactly once.

OUTPUT

The first and the only output line should contain maximum sum of knowledge chosen students can have, with exactly one digit in decimal part.

EXAMPLE TEST DATA

```
input

3 2 2
2 3.0 1 0.2 3 0.1
3 1.0 2 0.5 1 0.2

output

4.0
```

```
input

4 4 3
4 5.0 2 4.0 3 2.0 1 1.0
2 2.0 3 1.0 1 0.5 4 0.3
4 6.0 3 5.0 2 2.0 1 0.0
1 4.0 2 3.0 4 0.6 3 0.3

output

15.0
```

Description of the first test case:
There are two categories. In the first category, best student is the second one, with knowledge 3.0. He is followed by student numbered 1, with knowledge 0.2, and then number 3, with knowledge 0.1.

Best solution is to choose students 2 and 3, in categories 1 and 2, respectively.

In this economy, we all know how hard it is to get a job. Mirko, a recent college graduate, however, got lucky - he is now employed as a runeologist by the Language Institute of Croatia. His friend Slavko believes runeology isn't a science and is hence angry at Mirko for believing the opposite. One foggy Christmas day, Mirko's laptop broke. Since he isn't great with computers, he gave it to Slavko to repair it. Slavko, felling a little naughty, decided to mess up a particular file Mirko was working on.

This file contains a **matrix** of **R** rows and **C** columns. Each element of the matrix is a single letter. **No two columns of the matrix are equal**. To have some fun with pseudo-scientist Mirko, Slavko decided he will **delete as much rows as possible** from the top of the table, without breaking the no-equal-column rule.

INPUT

The first line of input contains two integers **R** and **C** ($2 \le$ **R**, **C** ≤ 1000), the number of rows and the number of columns, respectively.

In each of the next **R** lines there are **C** small letters of the English alphabet. These **R** x **C** letters represent Mirko's table (which does not have two same columns).

OUTPUT

Output a single integer, the maximum number of rows which can be deleted from the top of the table so that no two columns are equal.

EXAMPLE TEST DATA

input 2 6 dobarz adatak **output** 0	**input** 3 4 alfa beta zeta **output** 2	**input** 4 6 mrvica mrvica marica mateja **output** 1

Mirko discovered what Slavko did in previous task, and decided to deal with something completely opposite to tables of letters: sequences of numbers.

Let's define a **value** of a sequence as the difference between the largest and the smallest number within that sequence. For example, value of sequence (3, 1, 7, 2) is 6, and value of (42, 42) is 0.

Find the **sum** of values of all subsequences of **consecutive elements** of a given sequence.

INPUT

The first line of input contains a single integer **N** (2 ≤ **N** ≤ 300 000), number of elements of the sequence.

Next **N** lines contain elements of the sequence. Each element is a positive integer not greater than 100 000 000.

OUTPUT

The first and only line of output must contain the requested sum.

EXAMPLE TEST DATA

input 3 1 2 3	**input** 4 7 5 7 5	**input** 4 3 1 7 2
output 4	**output** 12	**output** 31

Mirko soon realised that number sequences are not the best career choice, and went right back to letter-table business.

Mirko's table has **R** rows and **C** columns and consists of lowercase letters.

Each cell of the table is a square of equal size. We assign coordinates to vertices of those squares, so that upper-left corner of the table has coordinates (0, 0), upper-right (**C**,0), lower-left (0, **R**), and lower-right (**C**, **R**).

We say that **polygon** within the table is **monoliteral** if the following holds:

1. its vertices are from the described set of cell-square vertices,
2. its edges are parallel to coordinate axes,
3. all letters inside the polygon are equal.

A simple polygon for which first two conditions are true (third one may or may not be true) is given. Mirko would like to know the number of **monoliteral** polygons that can be obtained by **moving** the given one up, down, left, or right or any combination thereof, but **not rotating**.

INPUT

The first line of input contains two space seperated integers **R** and **C** ($1 \le$ **R**, **C** ≤ 500).

Each of the next **R** lines contains exactly **C** lowercase letters, this is Mirko's table.

The following line contains integer **V** ($4 \le$ **V** ≤ 500), number of vertices of given polygon.

Each of the next **V** lines contains two integers **X**, **Y** ($0 \le$ **X** $\le$ **C**, $0 \le$ **Y** $\le$ **R**). These are the coordinates of the vertices of the given polygon. Vertices are given in clockwise order.

The given polygon will satisfy conditions 1 and 2 from above.

OUTPUT

In the first and only line of output, print expected number of polygons.

SCORING

In test cases worth 40% of total points, **R**, **C** and **V** will not exceed 20.

In test cases worth 70% of total points, **V** will not exceed 20.

EXAMPLE TEST DATA

input 3 3 aaa aaa aaa 4 2 0 2 2 0 2 0 0	**input** 3 3 aaa aba aaa 4 2 0 2 2 0 2 0 0	**input** 5 4 xyyx xyyy xxyy xxxx xxxx 8 1 3 1 2 0 2 0 0 2 0 2 1 3 1 3 3
output 4	**output** 0	**output** 2