South African Computer Olympiad
Final Round 2011
Day 1

Overview

Author(s)			
Problem	**squares**	**anagrams**	**language**
Source	squares.java squares.py squares.c squares.cpp squares.pas	anagrams.java anagrams.py anagrams.c anagrams.cpp anagrams.pas	language.java language.py language.c language.cpp language.pas
Input file	stdin	stdin	stdin
Output file	stdout	stdout	stdout
Time limit	2 seconds	0.7 seconds	0.7 seconds
Memory limit	64MiB	64MiB	64MiB
Number of tests	10	10	10
Points per test	10	10	10
Detailed feedback	Yes	No	No
Total points	**100**	**100**	**100**

The maximum total score is 300 points.

Sat 03 Sep 2011

Finding Squares

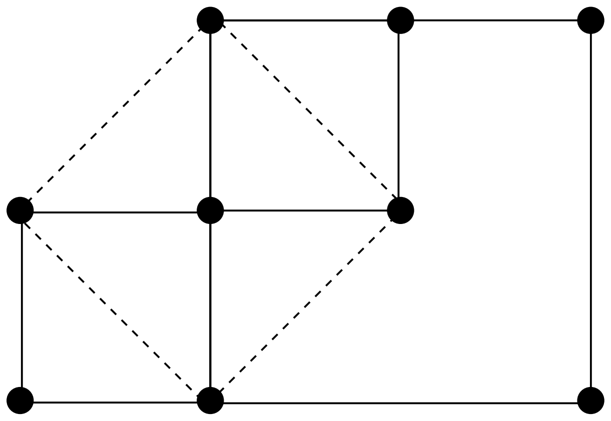
Introduction

Fred the Manic Storekeeper likes squares, especially ones whose sides go due North-South and East-West. He calls these axis-aligned squares. He wants to know how many such squares are formed by the items in his warehouse.

Task

Given the positions of the N items in Fred's warehouse, count the number of axis-aligned squares which have an item at each corner.

Example



Fred has nine items in his warehouse. They are positioned as shown above. There are three axis-aligned squares (shown with solid lines). The dashed square is not axis-aligned.

Input (stdin)

The first line of input contains a single integer, N. The next N lines describe the N items: each line contains two space-separated integers, x and y, the coordinates of the item. Each item is at different coordinates.

Sample input

```
9
0 0
1 0
0 1
1 1
1 2
2 1
2 2
3 2
3 0
```

Output (stdout)

The output consists of a single line containing a single integer, the number of axis-aligned squares.

Sample output

```
3
```

Constraints

- $1 \leq N \leq 2\,000$
- $-1\,000\,000 \leq x, y \leq 1\,000\,000$

Additionally, in 50% of the test cases:

- $1 \leq N \leq 30$

Time limit

2 seconds.

Detailed feedback

Detailed feedback is enabled for this problem.

Scoring

A correct solution will score 100% while an incorrect solution will score 0%.

Anagram Escape

Introduction

Bruce has been imprisoned by the Chinese IOI team. Carl is planning to help him escape. He needs to let Bruce know when he is coming, but all letters between them are closely monitored by the prison guards.

Fortunately, they were able to arrange a system to communicate before Bruce was imprisoned. Bruce and Carl have decided on a certain string, which they call the key. Carl will send Bruce a letter which appears innocent, but contains a hidden message indicating the date Carl will break into the prison. If Carl plans to break in D days after sending the letter, he will include the key D times in the letter.

However, if the same string appears too many times in a letter, the prison guards may become suspicious. Therefore, Carl will use anagrams of the key.

An anagram of a string is a string with the same characters in a possibly different order. For example, “listen” is an anagram of “silent”, but “listens” is not since there is only one “s” in “silent”. Any word is an anagram of itself.

The guards also require that letters be written in Chinese so that they can understand them. For ease of programming, we will replace each Chinese character with an integer.

Task

Bruce is worried that he will miss some of the anagrams and get the wrong day, so he wants a program to check against.

Write a program which, given the key and the text of the letter, determines the number of times, D, that an anagram of the key appears in the letter. Appearances of the key may overlap.

Example

Suppose the key is “93 37 93”, and the message reads “56 93 93 37 93 42 37”.

The message contains two anagrams of the key: “93 93 37” and “93 37 93”. Thus Carl will arrive in two days' time.

Input (stdin)

The input consists of three lines. The first line contains two integers: K, the number of characters in the key, and N, the number of characters in the letter. The second line contains K integers, corresponding to the characters in the key. The third line contains N integers, corresponding to the characters in the letter.

Sample input

```
3 7
93 37 93
56 93 93 37 93 42 37
```

Output (stdout)

The output consists of a single line containing a single integer, D.

Sample output

```
2
```

Constraints

- $1 \leq K \leq N \leq 100\,000$
- $0 \leq$ each character number $< 20\,000$

Additionally, in 50% of the test cases:

- each character number < 100

Additionally, in 30% of the test cases:

- $1 \leq K \leq N \leq 1\,000$

Time limit

0.7 seconds.

Scoring

A correct solution will score 100% while an incorrect solution will score 0%.

Language Chains

Introduction

The work of the United Nations (UN) involves speakers of hundreds of different languages, so they employ many translators to translate documents between languages.

To reduce the number of translators needed, they allow people to translate between any two languages they know, and sometimes use "chain translation" where one or more intermediate languages is used.

Task

The UN wants to ensure that they can translate documents between any pair of languages, possibly training some of their existing translators (numbered 1 to N) in additional languages. The languages are numbered 1 to L.

Determine which languages should be taught to which translators to achieve this goal. To minimize the cost of training, do so in such a way that minimizes the total number of additional language-translator pairs.

Example

Suppose there are four translators, who speak the following languages:

1. English and Chinese
2. Arabic and Chinese
3. Czech, Slovak and Russian
4. Hindi

Documents can be translated from English to Arabic: translator 1 translates from English to Chinese and then translator 2 translates from Chinese to Arabic. On the other hand, documents cannot be translated from Slovak to Chinese.

There are several ways to ensure documents can be translated between all eight of these languages. The best of these use only one additional translator-language pair. One option is to train translator 1 in Czech and Hindi.

Input (stdin)

The first line of input contains two integers, N and L. The second line contains a single integer, the current number of language-translator pairs, P. The next P lines each contain two integers, n and l, indicating that translator number n knows language number l.

Sample input

```
4 7
8
4 7
1 1
1 2
2 3
2 2
3 4
3 5
3 6
```

Output (stdout)

The output consists of zero or more lines each containing two integers, n and l, indicating that translator number n should be trained in language number l. If there are multiple correct answers, output any of them.

Sample output

```
1 4
1 7
```

Constraints

- $1 \leq l \leq L \leq 10\,000$
- $1 \leq n \leq N \leq 10\,000$
- $0 \leq P \leq 20\,000$

Additionally, in 50% of the test cases:

- $1 \leq L \leq 1\,000$
- $1 \leq N \leq 1\,000$
- $0 \leq P \leq 1\,000$

Time limit

0.7 seconds.

Scoring

A correct solution will score 100% while an incorrect solution will score 0%.