South African Computer Olympiad
Final Round
Day 1: Open

Overview

Author	**Bruce Merry**	**BIO 2002**	**USAICO 2006**
Problem	**joke**	**mouse**	**wooden**
Source	joke.java joke.py joke.c joke.cpp joke.pas	mouse.java mouse.py mouse.c mouse.cpp mouse.pas	wooden.java wooden.py wooden.c wooden.cpp wooden.pas
Input file	joke.in	mouse.in	wooden.in
Output file	joke.out	mouse.out	wooden.out
Time limit	1 second	1 second	1 second
Number of tests	10	10	10
Points per test	10	10	10
Total points	**100**	**100**	**100**

The maximum total score is 300 points.

Funniest Joke

Author

Bruce Merry

Introduction

Ernest Scribbler has written the funniest joke in the world. The joke is so funny that anyone who reads it dies almost instantly. The British army soon noticed its potential as a weapon and translated it into German to use against the Nazis. They will transmit the joke to their ground troops in an encrypted form to prevent it from being read accidentally and to keep it secret from Nazi spies. Help them check whether their encryption algorithm is reasonable by writing a program to perform letter frequency counts on the encrypted messages.

Task

The algorithm is to take each character of the original message and replace it with some string of characters (depending on the original character). Since keeping the joke a secret is so important, they may do this multiple times.

Your program will be given an original message, and a character range. It must return the frequency of each letter in the given range of the encrypted string.

Example

Suppose the original message is "badbeef". The replacement table is as follows:

$a \to dc$	$b \to aa$
$c \to facb$	$d \to a$
$e \to ea$	$f \to f$

After one iteration, the message would be "aadcaaaeaeaf". After a second iteration it becomes "dcdcafacbdcdcdceadceadcf". The frequencies of letters between positions 4 and 15 (underlined) are 2 a's, 1 b, 5 c's, 3 d's and 1 f.

Input (joke.in)

The first line contains four integers separated by spaces, L, A, B and N. Only the first L letters of the alphabet are used in both the original messages and the replacement table. You are asked for the frequencies between positions A and B in the encryption (inclusive, and counting from 1). N is the number of iterations of the encryption algorithm.

The next L lines each contain a non-empty string of lower-case letters. The first is the replacement string for "a", the second for "b" and so on. The following line contains another non-empty string of lower-case letters, which is the message that is going to be encrypted.

Sample input

```
6 4 15 2
dc
aa
facb
a
ea
f
badbeef
```

Output (joke.out)

The output consists of L integers, one per line. The first is the number of a's in the range, the second the number of b's and so on.

Sample output

```
2
1
5
3
0
1
```

Constraints

- $1 \leq A \leq B < 2^{30}$
- $1 \leq L \leq 26$
- $1 \leq N \leq 20$
- $1 \leq$ length of each replacement string ≤ 4
- $1 \leq$ length of the original message ≤ 10

In 50% of the test cases, $B - A \leq 10^6$.

Time limit

1 second. Python: 10 seconds.

The Mouse Problem

Author

BIO 2002

Introduction

The police have started to infiltrate the mouse parties, dressing up as voles so that they can identify closet mice. They have been quite successful so far because few of the mice know each other, and so the infiltrators do not obviously stand out. Help the mice to get to know each other better, so that they cannot be so easily infiltrated.

Task

The mice are going to stand in a circle and eat cheese. The goal is to have each mouse stand between two other mice that he/she does not know, to whom he/she can then be introduced. Fortunately this will be relatively easy to arrange, because each mouse knows less than half of the other mice.

Example

Suppose there are 6 mice, numbered 1 to 6. The pairs (1, 2), (2, 3), (3, 4), (4, 5), (5, 6) and (6, 1) all know each other (it is assumed that if A knows B then B knows A). If the mice stand in the order 1, 4, 2, 6, 3, 5 around the circle, then no pair of neighbouring mice will know each other.

Input (mouse.in)

The first line of input contains the integers N and F, separated by a space. N is the number of mice (numbered from 1 to N) and F is the number of pairs that know each other. The following F lines each contain two integers A and B separated by a space, indicating that mice A and B know each other. It is guaranteed that $A \neq B$ and that no pair is listed twice.

Sample input

```
6 6
1 2
2 3
3 4
4 5
5 6
6 1
```

Output (mouse.out)

The output consists of N lines, each of which is the number of a mouse. The numbers indicate the order in which the mice stand in the circle.

Sample output

```
1
4
2
6
3
5
```

Constraints

- $3 \leq N \leq 1000$

In 50% of the test cases, $N \leq 20$.

Time limit

1 second. Python: 10 seconds.

Wooden animals

Author

USAICO 2006

Introduction

Sir Bedevere's plan of using a giant wooden rabbit to infiltrate the French castle failed, as did his plan of using a giant wooden badger. He has continued sending giant wooden animals into the castle, and the French now have more than they know what to do with. They have found an unused barn, which they wish to fill as a much as possible with the animals they have.

Task

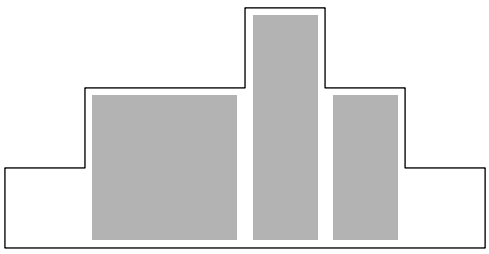
You will be given the shape of the barn and the sizes of the animals. Determine the maximum number of animals that can be placed in the barn.

The animals all have the same length, which (rather conveniently) is the depth of the barn. However, the animals have varying widths and heights. The roof of the barn is somewhat unusual: rather than sloping, it consists of vertical and horizontal panels. When walking from one side of the barn to the other, the roof first gets only higher and higher, and after reaching the highest point gets only lower and lower until reaching the other side.

The animals are on wheels, so they must not be stacked or tipped over. For this problem, treat each animal as a rectangle with a certain width and height. The goal is to find the maximum total area of the rectangles that are packed into the barn.

Example

Suppose there are four animals of sizes 4×1, 2×1, 2×2 and 1×3, with the barn as shown in the outline below. The grey blocks show the latter three animals placed into the barn, giving a total area of 9.



Input (wooden.in)

The first line of input contains two space-separated integers, N and W. N is the number of animals and W is the width of the barn. The next N lines each contain two space-separated integers, w_i and h_i, which are the width and height of one of the animals. The next W lines each contain one integer. The ith of these integers is the height of the roof between $i-1$ and i units from the left end of the barn.

Sample input

```
4 6
4 1
2 1
2 2
1 3
1
2
2
3
2
1
```

Output (wooden.out)

The output consists of a line containing a single integer, the maximum total area of the animals that are packed into the barn.

Sample output

```
9
```

Constraints

- $1 \leq N \leq 100$
- $1 \leq W \leq 100$
- $1 \leq$ height of roof ≤ 100
- $1 \leq w_i, h_i \leq 100$

In 50% of the test cases, $1 \leq N \leq 10$.

Time limit

1 second. Python: 10 seconds.