IOI 2011 Team Selection Test

Task	Program Name	Test Cases	Points per case	Total points	Memory limit	Time limit	Source Code limit
Word Search	word.c word.cpp word.pas	20	5	100	256 MB	3s	50000 bytes
Moon Map	map.c map.cpp map.pas	10	10	100	256 MB	3s	50000 bytes
Ticket Office	ticket.c ticket.cpp ticket.pas	20	5	100	256 MB	3s	50000 bytes

Word Search

Consider the following letter grid:

P	N	E	T
E	T	X	N
E	S	T	S

There are 7 ways to read the word TENTS from the grid:

P	$\underline{N}$	$\underline{E}$	$\underline{T}$
E	$\underline{T}$	X	N
E	$\underline{S}$	T	S

P	N	$\underline{E}$	T
E	$\underline{T}$	X	$\underline{N}$
E	$\underline{S}$	$\underline{T}$	S

P	$\underline{N}$	$\underline{E}$	T
E	$\underline{T}$	X	N
E	$\underline{S}$	T	S

P	N	$\underline{E}$	$\underline{T}$
E	T	X	$\underline{N}$
E	S	$\underline{T}$	$\underline{S}$

P	$\underline{N}$	E	T
$\underline{E}$	$\underline{T}$	X	N
E	$\underline{S}$	T	S

P	N	$\underline{E}$	T
E	$\underline{T}$	X	$\underline{N}$
E	S	$\underline{T}$	$\underline{S}$

P	N	$\underline{E}$	$\underline{T}$
E	T	X	$\underline{N}$
E	$\underline{S}$	$\underline{T}$	S

Given a letter grid and a word, your task is to determine the number of ways the word can be read from the grid. The first letter of the word can be in any cell of the grid, and after each letter, the next letter has to be in one of the neighbour cells (horizontally, vertically or diagonally). A cell can be used multiple times when reading the word.

Input

The first line of the input contains three integers: H ($1 \leq H \leq 200$), the height of the grid, W ($1 \leq W \leq 200$), the width of the grid, and L ($1 \leq L \leq 100$), the length of the word. The following H lines each containing W letters describe the grid. The last line containing L letters describes the word. All letters in the grid and in the word are uppercase English letters (A. . . Z).

Output

The only line of the output should contain one integer: the number of ways the word can be read from the grid. You may assume that the answer is always at most 10^{18}.

Sample

```
input         output
3 4 5         7
PNET
ETXN
ESTS
TENTS
```

Sample

```
input         output
2 2 10        78732
AA
AA
AAAAAAAAAA
```

Moon Map

In the year 2051, several Moon expeditions have explored different areas of the moon and produced maps of these areas. Now, the SLSA (Sri Lanka Space Agency) has an ambitious plan: they would like to produce a map of the whole planet. In order to calculate the necessary effort, they need to know the total size of the area for which maps already exist. It is your task to write a program that calculates this area.

Write a program that:

- reads the description of map shapes from the input
- computes the total area covered by the maps,
- writes the result to the output

Input

The input starts with a line containing a single integer *N* (*1<=N<=10 000*), the number of available maps. Each of the following *N* lines describes a map. Each of these lines contains four integers x_1, y_1, x_2 and y_2 ($0<=x_1<x_2<=30\ 000$, $0<=y_1<y_2<=30\ 000$). The values (x_1,y_1) and (x_2,y_2) are the coordinates of, respectively, the bottom-left and the top-right corner of the mapped area. Each map has rectangular shape, and its sides are parallel to the *x*- and *y*-axis of the coordinate system.

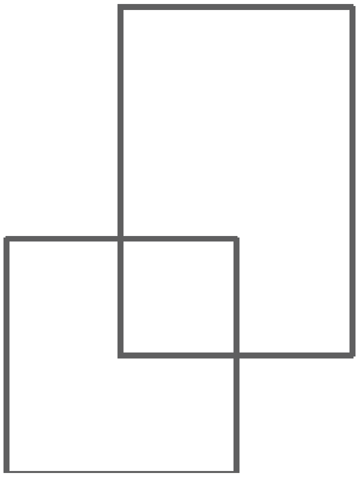
Output

The output should contain one integer *A*, the total explored area (i.e. the area of the union of all rectangles).

Example

Input

```
2
10 10 20 20
15 15 25 30
```



Correct output

```
225
```

Ticket Office

A ticket office sells tickets for concerts. Instead of selling tickets for single seats it sells bunches of tickets for a fixed number of consecutive seats. The office has received a great number of purchase orders. A purchase order for one bunch of seats specifies the lowest seat number (i.e. position of the first desired seat) in the bunch.

The office may not be able to fulfill all the purchase orders. Moreover, if it only allocates seats exactly as requested in the purchase orders then a great number of seats could remain empty. Therefore, the office applies the following allocation and pricing strategy. If a purchase order is accepted and the allocated seats are exactly those that are requested then the purchaser pays full-price (Rs. 2 for the bunch). If a purchase order is accepted, but the allocated seats differ from the requested ones (by even one position) then the purchaser pays half-price (Rs. 1 for the bunch). Of course it is desirable to maximize the total income.

You are to write a program that computes the maximal income that can be achieved and allocate the seats to the selected orders achieving this income.

Input

The first line of the input contains two integers, M and L. M ($1 \le M \le 30\,000$) is the number of seats and L ($1 \le L \le 100$) is the number of consecutive seats in every bunch. Seats are numbered from **1** to M. The second line contains an integer, N ($1 \le N \le 100\,000$), the number of purchase orders. The third line contains N integers, defining the purchase orders. The ith number in the line, z ($1 \le z \le M\text{-}L+1$), means that the ith purchaser requests the bunch of seats starting at seat z and ending at seat $z+L-1$.

Output

The first line of the output contains an integer, S, the maximal income. The second line contains an integer, Q, the number of accepted orders. The next Q lines describe the seat allocations. Each line contains a pair of integers, $x\ y$. The pair $x\ y$ means that the purchaser x gets the seats starting at seat number y. The lines must be written in the increasing order of seat numbers.

If there are multiple possibilities, your program should output only one; it does not matter which one.

Example

Input	output

20 3
7
4 2 10 9 16 15 17
	9
	6
	4 1
	1 4
	2 7
	3 10
	6 13
	5 16