problem	broj	fusnote	haker	hugo

source file	`broj.pas`
	`broj.c`
	`broj.cpp`
		`fusnote.pas`
		`fusnote.c`
		`fusnote.cpp`
			`haker.pas`
			`haker.c`
			`haker.cpp`
				`hugo.pas`
				`hugo.c`
				`hugo.cpp`

input data	stdin			
output data	stdout			
memory limit	16 MB			
time limit (pentium4 2.4 ghz)	5 seconds			
points	**30**	**40**	**60**	**70**
	200			

A sequence of digits is obtained by writing down decimal representations of **all** integers starting with **1** and continuing up to a certain number **N** consecutively like this:

12345678910111213141516171819202122 ... etc.

Write a program that will compute **the number of digits** in such a sequence.

input data

The first and only line of input contains an integer **N**, $1 \leq \mathbf{N} \leq 100{,}000{,}000$.

output data

The first and only line of output should contain the number of digits in a sequence determined by the number given in the input.

examples

input	input	input
5	15	120
output	**output**	**output**
5	21	252

A text in a book consists of a sequence of lines. A line may contain references to **footnotes**. A footnote consists of one or more lines and it have to be printed **together with its reference** on the same page. Once a footnote is printed on a page, only another footnote may follow it on that page. A **maximal number of lines** that can be printed on one page is known. No page of a book may contain more than that number of lines, including footnotes.

Write a program that will compute the **minimal** number of pages a book can have.

input data

The first line of input contains two integers: N, a number of lines in a document ($2 \leq N \leq 1000$), and K, maximal number of lines a page of a book may contain ($2 \leq K \leq 1000$), separated by a space character.

The second line of input contains an integer F, $1 \leq F \leq 100$, a number of footnotes in a book.

Each of the next F lines consists of two numbers, X and Y, separated by a space character, meaning that X-th line of the text has a reference to a footnote consisting of Y lines. Those footnotes descriptions will be **sorted** with respect to the lines where they are being referenced.

Note: Input data will be chosen so that a solution always exists.

output data

The first and only line of output should contain the minimal number of pages a book can have.

examples

input
```
5 5
1
3 2
```

output
```
2
```

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

output
```
4
```

input
```
10 5
5
3 3
4 1
6 2
6 1
9 3
```

output
```
6
```

Charlie is a skilful Internet surfer and he can prove it by many e-mail addresses he uses regularly. Their use is password protected and he, not having good memory, had to think of a simple method of making and remembering them. A passwords Charlie use for his e-mail protection consist of **two names of girls** he secretly adores **strung together**.

He secretly adores **at least two** and **at most five** girls, they all have **different** names, and their names consist of **at least three** and **at most eight** letters. Lucy knows how Charlie constructs his e-mail passwords, somehow found them all and is eager to find the names of girls Charlie secretly adores. Write a program that will help Lucy to find **the minimal set** of names Charlie used to make his e-mail passwords.

input data

The first line of input contains an integer N, a number of passwords ($1 \leq N \leq 100$).

The next N lines of the input contain passwords. A password is a sequence of at most 16 characters 'a'-'z'.

output data

The first line of output should contain number S, the minimal number of names the passwords are made of.

The names should be contained in the next S lines of the output, **sorted** in ascending order.

Note: Input data will be chosen so that the solution will be unique.

examples

input
```
2
ivaana
anaiva
```
output
```
2
ana
iva
```

input
```
3
ananana
nanahana
hanaana
```
output
```
3
ana
hana
nana
```

input
```
3
nemikirk
daglaskirk
kirkdaglas
```
output
```
3
daglas
kirk
nemi
```

hugo

Hugo is a PC game hero that has to collect as many apples falling down the trees as possible. He walks, or runs, at the bottom of a screen where a path divided into N equal squares is drawn. The squares are labelled by the numbers from **1** (the **leftmost** square) to **N** (the **rightmost** square). There is an apple tree rising above each square. Every now and then an apple falls down from a tree on a screen.

At the beginning of a game Hugo is positioned at the **middle** square (there are **odd** number of squares). If Hugo is positioned at the square P, then the **next** second he can move to the **left** to the square P-1, or to the **right** to the square P+1, or he may choose to **stay** at the square P. He stays at a square one second or more. If Hugo is positioned at the right square he will catch the falling apple. He knows in advance when an apple will fall and from which tree.

Write a program that will compute the **maximal** number of apples Hugo can catch.

input data

The first line of input contains an odd integer N, $1 \leq N \leq 999$, the number of squares. The next N lines contain data about falling apples.

Data concerning M-th tree are contained in the $(M+1)^{st}$ line. Each of those lines contains an integer number K, $1 \leq K \leq 3000$, followed by K **ascending** integers. Those numbers denotes times when apples will fall from the tree. The largest time would be less than or equal to 100,000.

output data

The first and only line of output should contain the maximal number of apples that can be caught.

examples

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

output
```
4
```

input
```
7
2 1 2
1 8
2 2 3
2 2 5
2 1 2
2 3 5
2 3 7
```

output
```
4
```

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

output
```
7
```