

---

---

## problems

---

---

problem	gus	niz	gumbi	piloti
source file	gus.pas gus.c gus.cpp	niz.pas niz.c niz.cpp	gumbi.pas gumbi.c gumbi.cpp	piloti.pas piloti.c piloti.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			

## gus

---

A worm is going to eat chocolate in a form of a **rectangle** with **M rows** and **N columns**.

It starts eating at the **upper left** corner and eats the entire row to its end. It then turns **clockwise** and **continues** to eat the whole column (the last one). Upon reaching its end, the process is repeated, i.e. the worm turns clockwise and eats the entire last row. By repeating this process of eating the chocolate, worm will eat the whole chocolate.

Write a program that will compute **the number of turns** the worm made until it ate the whole chocolate.

### input data

The first and only line of the input contains two integers, M and N,  $2 \leq M, N \leq 100$ ,. separated by a space character. M is number of rows and N is number of columns.

### output data

The first and only line of output should contain the number of turns a worm needs to make until it eats the whole chocolate.

### examples

**input**

2 3

**input**

5 3

**input**

4 10

**output**

2

**output**

5

**output**

6

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 **position of the first occurrence** of the decimal representation of number **N** in the sequence.

### input data

The first and only line of the input contains the integer  $N$ ,  $1 \leq N \leq 100,000$ .

### output data

The first and only line of output should contain the position of the first occurrence of the decimal representation of number  $N$  in the sequence.

### examples

**input**

15

**input**

34

**input**

142

**output**

20

**output**

3

**output**

73

## gumbi

---

Dave has a TV that is quite old, and some of its switches are not functioning properly. Pressing a switch (while TV was new) **released** all the other switches thus leaving only one switch in pressed position.

Pressing a switch now will release **some** of the switches leaving the positions of other switches **unchanged**. Dave knows the effects of pressing every switch.

Write a program that will help Dave determine **the length of the shortest sequence** of switches that need to be pressed from their given position so that at the end only switch 3 is in pressed position and all the others are released.

### input data

The first line of the input contains an integer  $N$ ,  $3 \leq N \leq 20$ , the number of TV switches.

The second line of input contains  $N$  binary digits separated by a space character determining the **initial position** of switches - 0 means that the corresponding switch is **released**, and 1 that it is in **pressed** position.

The next  $N$  lines of input determine the switches that are released by pressing a switch.  $(M+2)^{\text{th}}$  line begins with a number  $K$  followed by  $K$  numbers (sorted in ascending order) - switches that are released when the switch  $M$  is pressed. (Switches are denoted by a numbers from 1 to  $M$ .) A switch cannot release itself and it is possible that a switch does not release any switch at all.

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

### output data

The first and only line of output should contain the length of the shortest sequence of switches that need to be pressed so that at the end only switch 3 is in pressed position and all the others are released.

### examples

<b>input</b>	<b>input</b>	<b>input</b>
3	4	5
1 1 0	0 1 0 1	1 1 0 0 1
2 2 3	3 2 3 4	4 2 3 4 5
2 1 3	1 1	4 1 3 4 5
2 1 2	1 1	2 2 4
	0	0
<b>output</b>		4 1 2 3 4
1	<b>output</b>	
	2	<b>output</b>
		3

## piloti

---

Charlie acquired airline transport company and to stay in business he needs to lower the expenses by any means possible.

There are **N pilots** working for his company (N is even) and **N/2 plane crews** needs to be made. A plane crew consists of two pilots - a **captain** and his **assistant**. A captain **must be older** than his assistant. Each pilot has a contract granting him two possible salaries - one as a captain and the other as an assistant. A captain's salary is larger than assistant's for the same pilot. However, it is possible that an assistant has larger salary than his captain.

Write a program that will compute the **minimal** amount of money Charlie needs to give for the pilots' salaries if he decides to spend some time to make the optimal (i.e. the cheapest) arrangement of pilots in crews.

### input data

The first line of input contains integer N,  $2 \leq N \leq 10,000$ , N is even, the number of pilots working for the Charlie's company.

The next N lines of input contain pilots' salaries. The lines are **sorted** by pilot's age, the salaries of the youngest pilot are given the first. Each of those N lines contains two integers separated by a space character, X i Y,  $1 \leq Y < X \leq 100,000$ , a salary as a captain (X) and a salary as an assistant (Y).

### output data

The first and only line of output should contain the minimal amount of money Charlie needs to give for the pilots' salaries.

### examples

input	input	input
4	6	6
5000 3000	10000 7000	5000 3000
6000 2000	9000 3000	4000 1000
8000 1000	6000 4000	9000 7000
9000 6000	5000 1000	11000 5000
	9000 3000	7000 3000
<b>output</b>	8000 6000	8000 6000
19000		
	<b>output</b>	<b>output</b>
	32000	33000