

Problems

Problem	SAFE	PIGS
Source file	safe.pas safe.c safe.cpp	pigs.pas pigs.c pigs.cpp
Input file	safe.in	pigs.in
Output file	safe.out	pigs.out
Time limit per test	10 sekundi	10 sekundi
Number of tests	10	10
Points per test	6	9
Total points	60	90
	150	

SAFE

Mirko has decided to rob mini-safe owned by his son so that he could take away some football stickers which are missing in his big sticker collection.

Lock on the safe consists of N **equal** discs, each of them being divided in 10,000,000 equal **segments** numbered from 1 to 10,000,000 clockwise. In the beginning, segments on discs having the same number are placed one above (beneath) the other. Discs are layed one on the other such that the segments are overlapping and each disc is **missing** exactly one segment that is called **hole**.

In order to unlock the lock, all holes must be one above (beneath) the other.

In **one** second, Mirko can turn one disc in **one** direction (clockwise or counterclockwise) for **one** segment.

Write a program which will find the **minimal** time that Mirko needs to open the safe.

Input data

The first line of the input file contains an integer N , $2 \leq N \leq 100,000$, the number of discs.

The next N lines contain data describing the initial positions of the holes on each disc.

The $(i + 1)$ -th line contains an integer P_i , $1 \leq P_i \leq 10000000$, the initial position of the hole on the i -th disc.

Output data

The first and only line of the output file should contain the minimal time (in seconds).

Note: take care of the size of that number.

Examples

safe.in

3
5
17
7

safe.out

12

safe.in

4
9999999
7
16
9999995

safe.out

29

safe.in

4
1
2500001
5000000
7500002

safe.out

9999998

PIGS

Mirko works on a pig farm that consists of M **locked** pig-houses and Mirko **can't** unlock any pig-house because he doesn't have the keys. Customers come to the farm one after another. Each of them has keys to some pig-houses and wants to buy a certain number of pigs.

All data concerning customers planning to visit the farm on that particular day are available to Mirko early in the morning so that he can make a sales-plan in order to **maximize** the number of pigs sold.

More precisely, the procedure is as following: the customer arrives, opens all pig-houses to which he has the key, Mirko sells a certain number of pigs from all the unlocked pig-houses to him, and, if Mirko wants, he **can redistribute** the remaining pigs across the unlocked pig-houses.

An **unlimited** number of pigs can be placed in every pig-house.

Write a program that will find the **maximum** number of pigs that he can sell on that day.

Input data

The first line of input file contains two integers M and N , $1 \leq M \leq 1000$, $1 \leq N \leq 100$, number of pig-houses and number of customers. Pig houses are numbered from 1 to M and customers are numbered from 1 to N .

The next line contains M integers, for each pig-house initial number of pigs. The number of pigs in each pig-house is greater or equal to 0 and less or equal to 1000.

The next N lines contains records about the customers in the following form (record about the i -th customer is written in the $(i+2)$ -th line):

$A \ K_1 \ K_2 \ \dots \ K_A \ B$

It means that this customer has key to the pig-houses marked with the numbers K_1, K_2, \dots, K_A (sorted nondecreasingly) and that he wants to buy B pigs. Numbers A and B can be equal to 0.

Output data

The first and only line of the output file should contain the number of sold pigs.

Examples

pigs.in

```
3 3
3 1 10
2 1 2 2
2 1 3 3
1 2 6
```

pigs.out

7

pigs.in

```
6 6
6 3 2 0 1 3
2 1 2 0
1 3 3
1 1 1
2 2 3 8
2 4 5 2
2 4 6 6
```

pigs.out

15

pigs.in

```
11 5
1 2 2 1 0 2 4 1 1 1 2
5 1 2 3 4 5 3
4 1 2 6 7 5
2 3 8 1
3 3 6 11 5
3 8 9 10 3
```

pigs.out

17