

Problems

Problem	IVICA	KRUG	LUNA	BRODOVI
Executable file	ivica.exe	krug.exe	luna.exe	brodovi.exe
Source file	ivica.pas ivica.c ivica.cpp	krug.pas krug.c krug.cpp	luna.pas luna.c luna.cpp	brodovi.pas brodovi.c brodovi.cpp
Input file	ivica.in	krug.in	luna.in	brodovi.in
Output file	ivica.out	krug.out	luna.out	brodovi.out
Time limit per test	5 seconds	5 seconds	5 seconds	5 seconds
Number of tests	10	10	10	10
Points per test	3	4	6	7
Total points	30	40	60	70
	200			

IVICA

Since Croatia is becoming a skiing superpower, we have learned all there is about skiing. Nevertheless, let's remind what a ski race looks like. The race consists of **two** runs.

In the first run we have N skiers each with his own starting number between 1 and N . The skiers race one by one down the hill in the increasing order of their starting numbers. In this task, we are given the **current position** of each skier when reaching the finish line (i.e. in relation to the skiers who performed before him).

Only the M best placed skiers in the first run qualify for the second run. This time they go down the hill in the descending order of their finishing times of the first run (i.e. from the one ranked M -th to the one ranked first). In this task, similarly to the first run, we are also given the **current position** of each skier when reaching the finish line (combined, i.e. for both runs).

Write a program that will determine which skiers have won medals. We know that every skier has finished the race and that there are no skiers with the same finishing times (in both of the runs).

Input data

First line of the input file consists of 2 integers N and M separated by a whitespace, $3 \leq M \leq N \leq 100$.

Next N lines contain current positions of skiers competing in the first run, whilst the M lines coming after that contain current positions of skiers for both runs combined. Positions are given in the order in which skiers go down the hill.

Output data

Your program should output the starting number of the skier winning the gold medal to the first line of the output file, the starting number of the one winning silver to the second line and the starting number of the skier winning bronze to the third line.

Examples

ivica.in

```
3 3
1
2
3
1
1
1
```

ivica.out

```
1
2
3
```

ivica.in

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

ivica.out

```
2
3
4
```

ivica.in

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

ivica.out

```
5
3
2
```

KRUG

Mirko is playing an extremely interesting game.

He has a deck of N cards (numbered 1 to N) such that **no two cards have the same number**. The deck has been **shuffled**.

Mirko starts with the first card and traverses through the deck until reaching the card with **number 1** which he then **removes** from the deck. After that he searches and removes each of the cards numbered 2, 3, 4, etc. **always** starting from the place where he has found the previously removed card. Each time he **reaches the end of the deck he claps** his hands and starts again from the beginning of the deck. This procedure will eventually empty the deck and once the last card (the one numbered N) has been removed the game is over.

Write a program that, given an initial shuffling of deck, finds the number of times Mirko will clap his hands during the game.

Input data

First line of the input file contains an integer N , the number of cards in the deck, $1 \leq N \leq 100000$.

The next N lines contain numbers of cards given in order of their appearance in the initial shuffling.

Output data

The only line of the output file should contain the requested number of claps as defined above.

Examples

krug.in

3
2
1
3

krug.out

1

krug.in

5
3
5
1
4
2

krug.out

2

krug.in

7
3
6
7
1
5
4
2

krug.out

3

LUNA

There are M cars on a ride in an amusement park, numbered 1 to M , and N children are waiting in a queue to enter them.

Each car has its ride duration (**maximum 15 minutes**) and can host only **one child at a time**. Once a car is empty, the next child in queue enters it. If two cars are empty at the same time, the child enters the one having a **smaller** number.

Write a program that will calculate number of the car the last child in queue will enter.

Input data

The first line of the input file contains two integers N and M separated by a whitespace. N is the number of children in queue and M is number of cars on a ride, $1 \leq N \leq 2,000,000,000$; $1 \leq M \leq 1000$.

The next line consists of M integers separated by whitespaces. These integers represent durations of rides for each of M cars given in minutes. Each of these numbers will be not less than 1 and not greater than 15.

Output data

The first and the only line of the output file should contain the number of the car as defined above.

Examples

luna.in

```
3 5
7 8 9 7 8
```

luna.out

```
3
```

luna.in

```
7 2
3 2
```

luna.out

```
2
```

luna.in

```
22 5
1 2 3 4 5
```

luna.out

```
4
```

BRODOVI

During his childhood, Mirko liked to play “Sea battle” a lot, but once he got bored of it and invented a new game imaginatively called “Ships catch fish on a river”.

The game board consists of N fields lined up in a row and numbered 1 to N in ascending order from left to right. These fields represent a river on which M ships are to be placed. For each field **the amount of fish** swimming in that part of the river is known. Every ship has a specific length, i.e. it **occupies** the specific number of **consecutive** fields. A ship must somewhere drop an **anchor** but is allowed to do that in a certain field only. That means that one of the fields a ship will occupy is predetermined.

There can be only one ship or a part of only one ship on a single field of the board. We define the total amount of fish caught as the sum of amounts of fish swimming in all fields occupied by ships.

Goal of the game is to place **all** ships on the river in a way that the **total amount of fish caught is maximal**.

While playing an instance of this game, Mirko is in doubt whether the way he placed ships is optimal. Therefore he asked you to help him calculate the **maximum** possible amount of fish caught.

Input data

First line of the input file contains an integer N , the number of fields on board, $1 \leq N \leq 100000$.

In the next line there are N integers, separated by whitespaces, which represent the amounts of fish swimming in appropriate field, given in kilograms. These numbers will not be less than 1 nor greater than 100.

The next line contains an integer M , number of the ships, $1 \leq M \leq N$.

Each of the following M lines consists of two integers B and D separated by a whitespace. That means that the appropriate ship should drop the anchor in a field numbered with B and that its length is D .

Note: input data will always be such that the solution will exist

Output data

The only line of the output file should contain the maximum possible total amount of fish caught.

Examples

brodovi.in

```
11
2 5 3 4 7 6 2 1 3 8 5
2
8 3
3 2
```

brodovi.out

```
20
```

brodovi.in

```
13
3 2 4 7 2 1 3 6 1 2 6 4 1
2
5 7
11 4
```

brodovi.out

```
38
```

brodovi.in

```
11
1 1 6 4 4 1 1 3 10 1 1
3
2 3
6 4
10 2
```

brodovi.out

```
31
```