A. Elections

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Awruk is taking part in elections in his school. It is the final round. He has only one opponent — Elodreip. The are n students in the school. Each student has exactly k votes and is obligated to use all of them. So Awruk knows that if a person gives a_i votes for Elodreip, than he will get exactly $k - a_i$ votes from this person. Of course $0 \le k - a_i$ holds.

Awruk knows that if he loses his life is over. He has been speaking a lot with his friends and now he knows $a_1, a_2, \dots, a_n$ — how many votes for Elodreip each student wants to give. Now he wants to change the number k to win the elections. Of course he knows that bigger k means bigger chance that somebody may notice that he has changed something and then he will be disqualified.

So, Awruk knows $a_1, a_2, \dots, a_n$ — how many votes each student will give to his opponent. Help him select the smallest winning number k. In order to win, Awruk needs to get strictly more votes than Elodreip.

Input

The first line contains integer n ($1 \le n \le 100$) — the number of students in the school.

The second line contains n integers $a_1, a_2, \dots, a_n$ ($1 \le a_i \le 100$) — the number of votes each student gives to Elodreip.

Output

Output the smallest integer k ($k \ge \max a_i$) which gives Awruk the victory. In order to win, Awruk needs to get strictly more votes than Elodreip.

Examples

input

```
5
1 1 1 5 1
```

output

```
5
```

input

```
5
2 2 3 2 2
```

output

```
5
```

Note

In the first example, Elodreip gets $1 + 1 + 1 + 5 + 1 = 9$ votes. The smallest possible k is 5 (it surely can't be less due to the fourth person), and it leads to $4 + 4 + 4 + 0 + 4 = 16$ votes for Awruk, which is enough to win.

In the second example, Elodreip gets 11 votes. If $k = 4$, Awruk gets 9 votes and loses to Elodreip.

B. Lost Array

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Bajtek, known for his unusual gifts, recently got an integer array $x_0, x_1, \dots, x_{k-1}$.

Unfortunately, after a huge array-party with his extraordinary friends, he realized that he'd lost it. After hours spent on searching for a new toy, Bajtek found on the arrays producer's website another array a of length $n + 1$. As a formal description of a says, $a_0 = 0$ and for all other i ($1 \le i \le n$) $a_i = x_{(i-1) \bmod k} + a_{i-1}$, where $p \bmod q$ denotes the remainder of division p by q.

For example, if the $x = [1, 2, 3]$ and $n = 5$, then:

- $a_0 = 0$,
- $a_1 = x_{0 \bmod 3} + a_0 = x_0 + 0 = 1$,
- $a_2 = x_{1 \bmod 3} + a_1 = x_1 + 1 = 3$,
- $a_3 = x_{2 \bmod 3} + a_2 = x_2 + 3 = 6$,
- $a_4 = x_{3 \bmod 3} + a_3 = x_0 + 6 = 7$,

- $a_5 = x_{4 \bmod 3} + a_4 = x_1 + 7 = 9$.

So, if the $x = [1, 2, 3]$ and $n = 5$, then $a = [0, 1, 3, 6, 7, 9]$.

Now the boy hopes that he will be able to restore x from a! Knowing that $1 \le k \le n$, help him and find all possible values of k — possible lengths of the lost array.

Input

The first line contains exactly one integer n ($1 \le n \le 1000$) — the length of the array a, excluding the element a_0.

The second line contains n integers $a_1, a_2, \dots, a_n$ ($1 \le a_i \le 10^6$).

Note that a_0 is always 0 and is not given in the input.

Output

The first line of the output should contain one integer l denoting the number of correct lengths of the lost array.

The second line of the output should contain l integers — possible lengths of the lost array in **increasing** order.

Examples

input

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

output

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

input

```
5
1 3 5 6 8
```

output

```
2
3 5
```

input

```
3
1 5 3
```

output

```
1
3
```

Note

In the first example, any k is suitable, since a is an arithmetic progression.

Possible arrays x:

- $[1]$
- $[1, 1]$
- $[1, 1, 1]$
- $[1, 1, 1, 1]$
- $[1, 1, 1, 1, 1]$

In the second example, Bajtek's array can have three or five elements.

Possible arrays x:

- $[1, 2, 2]$
- $[1, 2, 2, 1, 2]$

For example, $k = 4$ is bad, since it leads to $6 + x_0 = 8$ and $0 + x_0 = 1$, which is an obvious contradiction.

In the third example, only $k = n$ is good.

Array $[1, 4, -2]$ satisfies the requirements.

Note that x_i may be negative.

C. Smallest Word

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

IA has so many colorful magnets on her fridge! Exactly one letter is written on each magnet, 'a' or 'b'. She loves to play with them, placing all magnets in a row. However, the girl is quickly bored and usually thinks how to make her entertainment more interesting.

Today, when IA looked at the fridge, she noticed that the word formed by magnets is really messy. "It would look much better when I'll swap some of them!" — thought the girl — "but how to do it?". After a while, she got an idea. IA will look at all prefixes with lengths from 1 to the length of the word and for each prefix she will either reverse this prefix or leave it as it is. She will consider the prefixes in the fixed order: from the shortest to the largest. She wants to get the lexicographically smallest possible word after she considers all prefixes. Can you help her, telling which prefixes should be chosen for reversing?

A string a is lexicographically smaller than a string b if and only if one of the following holds:

- a is a prefix of b, but $a \neq b$;
- in the first position where a and b differ, the string a has a letter that appears earlier in the alphabet than the corresponding letter in b.

Input

The first and the only line contains a string s ($1 \le |s| \le 1000$), describing the initial string formed by magnets. The string s consists only of characters 'a' and 'b'.

Output

Output exactly $|s|$ integers. If IA should reverse the i-th prefix (that is, the substring from 1 to i), the i-th integer should be equal to 1, and it should be equal to 0 otherwise.

If there are multiple possible sequences leading to the optimal answer, print any of them.

Examples

input
```
bbab
```
output
```
0 1 1 0
```

input
```
aaaaa
```
output
```
1 0 0 0 1
```

Note

In the first example, IA can reverse the second and the third prefix and get a string "abbb". She cannot get better result, since it is also lexicographically smallest string obtainable by permuting characters of the initial string.

In the second example, she can reverse any subset of prefixes — all letters are 'a'.

D. Mysterious Crime

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

Acingel is a small town. There was only one doctor here — Miss Ada. She was very friendly and nobody has ever said something bad about her, so who could've expected that Ada will be found dead in her house? Mr Gawry, world-famous detective, is appointed to find the criminal. He asked m neighbours of Ada about clients who have visited her in that unlucky day. Let's number the clients from 1 to n. Each neighbour's testimony is a permutation of these numbers, which describes the order in which clients have been seen by the asked neighbour.

However, some facts are very suspicious – how it is that, according to some of given permutations, some client has been seen in the morning, while in others he has been seen in the evening? "In the morning some of neighbours must have been sleeping!" — thinks Gawry — "and in the evening there's been too dark to see somebody's face...". Now he wants to delete some prefix and some suffix (both prefix and suffix can be empty) in each permutation, so that they'll be non-empty and equal to each other after that — some of the potential criminals may disappear, but the testimony won't stand in contradiction to each other.

In how many ways he can do it? Two ways are called different if the remaining common part is different.

Input

The first line contains two integers n and m ($1 \le n \le 100\,000$, $1 \le m \le 10$) — the number of suspects and the number of asked neighbors.

Each of the next m lines contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le n$). It is guaranteed that these integers form a correct permutation (that is, each number from 1 to n appears exactly once).

Output

Output a single integer denoting the number of ways to delete some prefix and some suffix of each permutation (possibly empty), such that the remaining parts will be equal and non-empty.

Examples

input

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

output

```
4
```

input

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

output

```
5
```

input

```
2 2
1 2
2 1
```

output

```
2
```

Note

In the first example, all possible common parts are $[1]$, $[2]$, $[3]$ and $[2, 3]$.

In the second and third examples, you can only leave common parts of length 1.

E. Train Hard, Win Easy

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Zibi is a competitive programming coach. There are n competitors who want to be prepared well. The training contests are quite unusual – there are two people in a team, two problems, and each competitor will code exactly one of them. Of course, people in one team will code different problems.

Rules of scoring also aren't typical. The first problem is always an implementation problem: you have to implement some well-known algorithm very fast and the time of your typing is rated. The second one is an awful geometry task and you just have to get it accepted in reasonable time. Here the length and difficulty of your code are important. After that, Zibi will give some penalty points (possibly negative) for each solution and the final score of the team is the sum of them (the **less** the score is, the better).

We know that the i-th competitor will always have score x_i when he codes the first task and y_i when he codes the second task. We can assume, that all competitors know each other's skills and during the contest distribute the problems in the way that minimizes their final score. Remember that each person codes exactly one problem in a contest.

Zibi wants all competitors to write a contest with each other. However, there are m pairs of people who really don't like to cooperate and they definitely won't write a contest together. Still, the coach is going to conduct trainings for all possible pairs of people, such that the people in pair don't hate each other. The coach is interested for each participant, what will be his or her sum of scores of all teams he trained in?

Input

The first line contains two integers n and m ($2 \leq n \leq 300\,000$, $0 \leq m \leq 300\,000$) — the number of participants and the number of pairs of people who will not write a contest together.

Each of the next n lines contains two integers x_i and y_i ($-10^9 \leq x_i, y_i \leq 10^9$) — the scores which will the i-th competitor get on the first problem and on the second problem. It is guaranteed that there are no two people having both x_i and y_i same.

Each of the next m lines contain two integers u_i and v_i ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$) — indices of people who don't want to write a contest in one team. Each unordered pair of indices will appear at most once.

Output

Output n integers — the sum of scores for all participants in the same order as they appear in the input.

Examples

input

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

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

output

```
3 0 3
```

input

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

output

```
0 0 0
```

input

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

output

```
4 14 4 16 10
```

Note

In the first example, there will be only one team consisting of persons 1 and 3. The optimal strategy for them is to assign the first task to the 3-rd person and the second task to the 1-st person, this will lead to score equal to $1+2=3$.

In the second example, nobody likes anyone, so there won't be any trainings. It seems that Zibi won't be titled coach in that case...

F. Make It One

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Janusz is a businessman. He owns a company "Januszex", which produces games for teenagers. Last hit of Januszex was a cool one-person game "Make it one". The player is given a sequence of n integers a_i.

It is allowed to select any subset of them, and the score is equal to the greatest common divisor of selected elements. The goal is to take as little elements as it is possible, getting the score 1. Now Janusz wonders, for given sequence, how much elements should the player choose?

Input

The first line contains an only integer n ($1 \le n \le 300\,000$) — the number of integers in the sequence.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 300\,000$).

Output

If there is no subset of the given sequence with gcd equal to 1, output -1.

Otherwise, output exactly one integer — the size of the smallest subset with gcd equal to 1.

Examples

input

```
3
10 6 15
```

output

```
3
```

input

```
3
2 4 6
```

output

```
-1
```

input
7 30 60 21 42 70 15 30
output
3

Note
In the first example, selecting a subset of all numbers gives a gcd of 1 and for all smaller subsets the gcd is greater than 1.

In the second example, for all subsets of numbers the gcd is at least 2.

G. Speckled Band

time limit per test: 7 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Ildar took a band (a thin strip of cloth) and colored it. Formally, the band has n cells, each of them is colored into one of 26 colors, so we can denote each color with one of the lowercase letters of English alphabet.

Ildar decided to take some segment of the band $[l, r]$ ($1 \le l \le r \le n$) he likes and cut it from the band. So he will create a new band that can be represented as a string $t = s_l s_{l+1} \ldots s_r$.

After that Ildar will play the following game: he cuts the band t into some new bands and counts the number of different bands among them. Formally, Ildar chooses $1 \le k \le |t|$ indexes $1 \le i_1 < i_2 < \ldots < i_k = |t|$ and cuts t to k bands-strings $t_1 t_2 \ldots t_{i_1}, t_{i_1+1} \ldots t_{i_2}, \ldots, t_{i_{k-1}+1} \ldots t_{i_k}$ and counts the number of different bands among them. He wants to know the minimal possible number of different bands he can get under the constraint that at least one band repeats at least two times. The result of the game is this number. If it is impossible to cut t in such a way, the result of the game is -1.

Unfortunately Ildar hasn't yet decided which segment he likes, but he has q segments-candidates $[l_1, r_1]$, $[l_2, r_2]$, ..., $[l_q, r_q]$. Your task is to calculate the result of the game for each of them.

Input
The first line contains one integer n ($1 \le n \le 200\,000$) — the length of the band Ildar has.

The second line contains a string s consisting of n lowercase English letters — the band Ildar has.

The third line contains a single integer q ($1 \le q \le 200\,000$) — the number of segments Ildar has chosen as candidates.

Each of the next q lines contains two integer integers l_i and r_i ($1 \le l_i \le r_i \le n$) denoting the ends of the i-th segment.

Output
Output q lines, where the i-th of them should contain the result of the game on the segment $[l_i, r_i]$.

Example

input
9 abcabcdce 7 1 6 4 7 5 9 6 9 1 9 3 6 4 4
output
1 -1 4 3 2 2 -1

Note
Consider the first example.

If Ildar chooses the segment $[1, 6]$, he cuts a string $t = abcabc$. If he cuts t into two bands abc and abc, the band abc repeats two times and the number of different tapes is 1. So, the result of this game is 1.

If Ildar chooses the segment $[4, 7]$, he cuts a string $t = abcd$. It is impossible to cut this band in such a way that there is at least one band repeating at least two times. So, the result of this game is -1.

If Ildar chooses the segment $[3, 6]$, he cuts a string $t = cabc$. If he cuts t into three bands c, ab and c, the band c repeats two times and the number of different bands is 2. So, the result of this game is 2.