A. Disturbed People

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

There is a house with n flats situated on the main street of Berlatov. Vova is watching this house every night. The house can be represented as an array of n integer numbers $a_1, a_2, \ldots, a_n$, where $a_i = 1$ if in the i-th flat the light is on and $a_i = 0$ otherwise.

Vova thinks that people in the i-th flats are disturbed and cannot sleep if and only if $1 < i < n$ and $a_{i-1} = a_{i+1} = 1$ and $a_i = 0$.

Vova is concerned by the following question: what is the minimum number k such that if people from exactly k pairwise distinct flats will turn off the lights then nobody will be disturbed? Your task is to find this number k.

Input

The first line of the input contains one integer n ($3 \le n \le 100$) — the number of flats in the house.

The second line of the input contains n integers $a_1, a_2, \ldots, a_n$ ($a_i \in \{0, 1\}$), where a_i is the state of light in the i-th flat.

Output

Print only one integer — the minimum number k such that if people from exactly k pairwise distinct flats will turn off the light then nobody will be disturbed.

Examples

input	Copy
10 1 1 0 1 1 0 1 0 1 0	
output	Copy
2	

input	Copy
5 1 1 0 0 0	
output	Copy
0	

input	Copy
4 1 1 1 1	
output	Copy
0	

Note

In the first example people from flats 2 and 7 or 4 and 7 can turn off the light and nobody will be disturbed. It can be shown that there is no better answer in this example.

There are no disturbed people in second and third examples.

B. Points on the line

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

We've got no test cases. A big olympiad is coming up. But the problemsetters' number one priority should be adding another problem to the round.

The **diameter** of a multiset of points on the line is the largest distance between two points from this set. For example, the diameter of the multiset $\{1, 3, 2, 1\}$ is 2.

Diameter of multiset consisting of one point is 0.

You are given n points on the line. What is the minimum number of points you have to remove, so that the diameter of the multiset of the remaining points will not exceed d?

Input

The first line contains two integers n and d ($1 \le n \le 100, 0 \le d \le 100$) — the amount of points and the maximum allowed diameter respectively.

The second line contains n space separated integers ($1 \le x_i \le 100$) — the coordinates of the points.

Output

Output a single integer — the minimum number of points you have to remove.

Examples

input	Copy
3 1 2 1 4	
output	Copy
1	

input	Copy
3 0 7 7 7	

output Copy

```
0
```

input Copy

```
6 3
1 3 4 6 9 10
```

output Copy

```
3
```

Note

In the first test case the optimal strategy is to remove the point with coordinate 4. The remaining points will have coordinates 1 and 2, so the diameter will be equal to $2 - 1 = 1$.

In the second test case the diameter is equal to 0, so its is unnecessary to remove any points.

In the third test case the optimal strategy is to remove points with coordinates 1, 9 and 10. The remaining points will have coordinates 3, 4 and 6, so the diameter will be equal to $6 - 3 = 3$.

C. Event Dates

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

On a history lesson the teacher asked Vasya to name the dates when n famous events took place. He doesn't remembers the exact dates but he remembers a segment of days $[l_i, r_i]$ (inclusive) on which the event could have taken place. However Vasya also remembers that there was at most one event in one day. Help him choose such n dates of famous events that will fulfill both conditions. It is guaranteed that it is possible.

Input

The first line contains one integer n ($1 \le n \le 100$) — the number of known events. Then follow n lines containing two integers l_i and r_i each ($1 \le l_i \le r_i \le 10^7$) — the earliest acceptable date and the latest acceptable date of the i-th event.

Output

Print n numbers — the dates on which the events took place. If there are several solutions, print any of them. It is guaranteed that a solution exists.

Examples

input Copy

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

output Copy

```
1 2 3
```

input Copy

```
2
1 3
1 3
```

output Copy

```
1 2
```

D. Non-zero Segments

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Kolya got an integer array $a_1, a_2, \ldots, a_n$. The array can contain both positive and negative integers, but Kolya doesn't like 0, so the array doesn't contain any zeros.

Kolya doesn't like that the sum of some subsegments of his array can be 0. The subsegment is some consecutive segment of elements of the array.

You have to help Kolya and change his array in such a way that it doesn't contain any subsegments with the sum 0. To reach this goal, you can insert any integers between any pair of adjacent elements of the array (integers can be really any: positive, negative, 0, any by absolute value, even such a huge that they can't be represented in most standard programming languages).

Your task is to find the minimum number of integers you have to insert into Kolya's array in such a way that the resulting array doesn't contain any subsegments with the sum 0.

Input

The first line of the input contains one integer n ($2 \le n \le 200\,000$) — the number of elements in Kolya's array.

The second line of the input contains n integers $a_1, a_2, \ldots, a_n$ ($-10^9 \le a_i \le 10^9, a_i \ne 0$) — the description of Kolya's array.

Output

Print the minimum number of integers you have to insert into Kolya's array in such a way that the resulting array doesn't contain any subsegments with the sum 0.

Examples

input Copy

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

output Copy

```
1
```

input Copy

```
5
4 -2 3 -9 2
```

output Copy

```
0
```

input Copy

```
9
-1 1 -1 1 -1 1 1 -1 -1
```

output Copy

```
6
```

input Copy

```
8
16 -5 -11 -15 10 5 4 -4
```

output Copy

```
3
```

Note

Consider the first example. There is only one subsegment with the sum 0. It starts in the second element and ends in the fourth element. It's enough to insert one element so the array doesn't contain any subsegments with the sum equal to zero. For example, it is possible to insert the integer 1 between second and third elements of the array.

There are no subsegments having sum 0 in the second example so you don't need to do anything.

E. Maximum Subsequence

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

You are given an array a consisting of n integers, and additionally an integer m. You have to choose some sequence of indices $b_1, b_2, ..., b_k$ ($1 \le b_1 < b_2 < ... < b_k \le n$) in such a way that the value of $\sum_{i=1}^{k} a_{b_i} \bmod m$ is maximized. Chosen sequence can be empty.

Print the maximum possible value of $\sum_{i=1}^{k} a_{b_i} \bmod m$.

Input

The first line contains two integers n and m ($1 \le n \le 35$, $1 \le m \le 10^9$).

The second line contains n integers $a_1, a_2, ..., a_n$ ($1 \le a_i \le 10^9$).

Output

Print the maximum possible value of $\sum_{i=1}^{k} a_{b_i} \bmod m$.

Examples

input Copy

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

output Copy

```
3
```

input Copy

```
3 20
199 41 299
```

output Copy

```
19
```

Note

In the first example you can choose a sequence $b = \{1, 2\}$, so the sum $\sum_{i=1}^{k} a_{b_i}$ is equal to 7 (and that's 3 after taking it modulo 4).

In the second example you can choose a sequence $b = \{3\}$.