Codeforces Round #171 (Div. 2)

A. Point on Spiral

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

Valera the horse lives on a plane. The Cartesian coordinate system is defined on this plane. Also an infinite spiral is painted on the plane. The spiral consists of segments: **[(0, 0), (1, 0)], [(1, 0), (1, 1)], [(1, 1), (- 1, 1)], [(- 1, 1), (- 1, - 1)], [(- 1, - 1), (2, - 1)], [(2, - 1), (2, 2)]** and so on. Thus, this infinite spiral passes through each integer point of the plane.

Valera the horse lives on the plane at coordinates $(0, 0)$. He wants to walk along the spiral to point (x, y). Valera the horse has four legs, so he finds turning very difficult. Count how many times he will have to turn if he goes along a spiral from point $(0, 0)$ to point (x, y).

Input

The first line contains two space-separated integers x and y $(|x|, |y| \leq 100)$.

Output

Print a single integer, showing how many times Valera has to turn.

Sample test(s)

input
0 0
output
0

input
1 0
output
0

input
0 1
output
2

input
-1 -1
output
3

B. Books

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

When Valera has got some free time, he goes to the library to read some books. Today he's got t free minutes to read. That's why Valera took n books in the library and for each book he estimated the time he is going to need to read it. Let's number the books by integers from 1 to n. Valera needs a_i minutes to read the i-th book.

Valera decided to choose an arbitrary book with number i and read the books one by one, starting from this book. In other words, he will first read book number i, then book number $i + 1$, then book number $i + 2$ and so on. He continues the process until he either runs out of the free time or finishes reading the n-th book. Valera reads each book up to the end, that is, he doesn't start reading the book if he doesn't have enough free time to finish reading it.

Print the maximum number of books Valera can read.

Input

The first line contains two integers n and t $(1 \le n \le 10^5; 1 \le t \le 10^9)$ — the number of books and the number of free minutes Valera's got. The second line contains a sequence of n integers $a_1, a_2, ..., a_n$ $(1 \le a_i \le 10^4)$, where number a_i shows the number of minutes that the boy needs to read the i-th book.

Output

Print a single integer — the maximum number of books Valera can read.

Sample test(s)

input
4 5 3 1 2 1
output
3

input
3 3 2 2 3
output
1

C. Ladder

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

You've got an array, consisting of n integers $a_1, a_2, ..., a_n$. Also, you've got m queries, the i-th query is described by two integers l_i, r_i. Numbers l_i, r_i define a subsegment of the original array, that is, the sequence of numbers $a_{l_i}, a_{l_i+1}, a_{l_i+2}, ..., a_{r_i}$. For each query you should check whether the corresponding segment is a ladder.

A *ladder* is a sequence of integers $b_1, b_2, ..., b_k$, such that it first doesn't decrease, then doesn't increase. In other words, there is such integer x $(1 \le x \le k)$, that the following inequation fulfills: $b_1 \le b_2 \le ... \le b_x \ge b_{x+1} \ge b_{x+2} ... \ge b_k$. Note that the non-decreasing and the non-increasing sequences are also considered ladders.

Input

The first line contains two integers n and m $(1 \le n, m \le 10^5)$ — the number of array elements and the number of queries. The second line contains the sequence of integers $a_1, a_2, ..., a_n$ $(1 \le a_i \le 10^9)$, where number a_i stands for the i-th array element.

The following m lines contain the description of the queries. The i-th line contains the description of the i-th query, consisting of two integers l_i, r_i $(1 \le l_i \le r_i \le n)$ — the boundaries of the subsegment of the initial array.

The numbers in the lines are separated by single spaces.

Output

Print m lines, in the i-th line print word "Yes" (without the quotes), if the subsegment that corresponds to the i-th query is the ladder, or word "No" (without the quotes) otherwise.

Sample test(s)

input
8 6 1 2 1 3 3 5 2 1 1 3 2 3 2 4 8 8 1 4 5 8

output
Yes Yes No Yes No Yes

D. The Minimum Number of Variables

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

You've got a positive integer sequence $a_1, a_2, ..., a_n$. All numbers in the sequence are distinct. Let's fix the set of variables $b_1, b_2, ..., b_m$. Initially each variable b_i $(1 \le i \le m)$ contains the value of zero. Consider the following sequence, consisting of n operations.

The first operation is assigning the value of a_1 to some variable b_x $(1 \le x \le m)$. Each of the following n - 1 operations is assigning to some variable b_y the value that is equal to the sum of values that are stored in the variables b_i and b_j $(1 \le i, j, y \le m)$. At that, the value that is assigned on the t-th operation, must equal a_t. For each operation numbers y, i, j are chosen anew.

Your task is to find the minimum number of variables m, such that those variables can help you perform the described sequence of operations.

Input

The first line contains integer n $(1 \le n \le 23)$. The second line contains n space-separated integers $a_1, a_2, ..., a_n$ $(1 \le a_k \le 10^9)$.

It is guaranteed that all numbers in the sequence are distinct.

Output

In a single line print a single number — the minimum number of variables m, such that those variables can help you perform the described sequence of operations.

If you cannot perform the sequence of operations at any m, print `-1`.

Sample test(s)

input
5 1 2 3 6 8
output
2

input
3 3 6 5
output
-1

input
6 2 4 8 6 10 18
output
3

Note

In the first sample, you can use two variables b_1 and b_2 to perform the following sequence of operations.

1. $b_1 := 1$;
2. $b_2 := b_1 + b_1$;
3. $b_1 := b_1 + b_2$;
4. $b_1 := b_1 + b_1$;
5. $b_1 := b_1 + b_2$.

E. Beautiful Decomposition

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

Valera considers a number *beautiful*, if it equals 2^k or -2^k for some integer k ($k \geq 0$). Recently, the math teacher asked Valera to represent number n as the sum of beautiful numbers. As Valera is really greedy, he wants to complete the task using as few beautiful numbers as possible.

Help Valera and find, how many numbers he is going to need. In other words, if you look at all decompositions of the number n into beautiful summands, you need to find the size of the decomposition which has the fewest summands.

Input

The first line contains string s ($1 \leq |s| \leq 10^6$), that is the binary representation of number n without leading zeroes ($n > 0$).

Output

Print a single integer — the minimum amount of beautiful numbers that give a total of n.

Sample test(s)

input
10
output
1

input
111
output
2

input
1101101
output
4

Note

In the first sample $n = 2$ is a beautiful number.

In the second sample $n = 7$ and Valera can decompose it into sum $2^3 + (-2^0)$.

In the third sample $n = 109$ can be decomposed into the sum of four summands as follows: $2^7 + (-2^4) + (-2^2) + 2^0$.