Codeforces Round #352 (Div. 2)

A. Summer Camp

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

Every year, hundreds of people come to summer camps, they learn new algorithms and solve hard problems.

This is your first year at summer camp, and you are asked to solve the following problem. All integers starting with 1 are written in one line. The prefix of these line is "`123456789101112131415...`". Your task is to print the n-th digit of this string (digits are numbered starting with 1.

Input

The only line of the input contains a single integer n $(1 \le n \le 1000)$ — the position of the digit you need to print.

Output

Print the n-th digit of the line.

Examples

input
3
output
3

input
11
output
0

Note

In the first sample the digit at position 3 is '`3`', as both integers 1 and 2 consist on one digit.

In the second sample, the digit at position 11 is '`0`', it belongs to the integer 10.

B. Different is Good

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

A wise man told Kerem "Different is good" once, so Kerem wants all things in his life to be different.

Kerem recently got a string s consisting of lowercase English letters. Since Kerem likes it when things are different, he wants all *substrings* of his string s to be distinct. Substring is a string formed by some number of consecutive characters of the string. For example, string "aba" has substrings "" (empty substring), "a", "b", "a", "ab", "ba", "aba".

If string s has at least two equal substrings then Kerem will change characters at some positions to some other lowercase English letters. Changing characters is a very tiring job, so Kerem want to perform as few changes as possible.

Your task is to find the minimum number of changes needed to make all the substrings of the given string distinct, or determine that it is impossible.

Input

The first line of the input contains an integer n ($1 \le n \le 100\,000$) — the length of the string s.

The second line contains the string s of length n consisting of only lowercase English letters.

Output

If it's impossible to change the string s such that all its substring are distinct print -1. Otherwise print the minimum required number of changes.

Examples

input
2 aa
output
1

input
4 koko
output
2

input
5 murat
output
0

Note

In the first sample one of the possible solutions is to change the first character to 'b'.

In the second sample, one may change the first character to 'a' and second character to 'b', so the string becomes "abko".

C. Recycling Bottles

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

It was recycling day in Kekoland. To celebrate it Adil and Bera went to Central Perk where they can take bottles from the ground and put them into a recycling bin.

We can think Central Perk as coordinate plane. There are n bottles on the ground, the i-th bottle is located at position (x_i, y_i). Both Adil and Bera can carry only one bottle at once each.

For both Adil and Bera the process looks as follows:

1. Choose to stop or to continue to collect bottles.
2. If the choice was to continue then choose some bottle and walk towards it.
3. Pick this bottle and walk to the recycling bin.
4. Go to step 1.

Adil and Bera may move independently. They are allowed to pick bottles simultaneously, all bottles may be picked by any of the two, it's allowed that one of them stays still while the other one continues to pick bottles.

They want to organize the process such that the total distance they walk (the sum of distance walked by Adil and distance walked by Bera) is minimum possible. Of course, at the end all bottles should lie in the recycling bin.

Input

First line of the input contains six integers a_x, a_y, b_x, b_y, t_x and t_y ($0 \le a_x, a_y, b_x, b_y, t_x, t_y \le 10^9$) — initial positions of Adil, Bera and recycling bin respectively.

The second line contains a single integer n ($1 \le n \le 100\,000$) — the number of bottles on the ground.

Then follow n lines, each of them contains two integers x_i and y_i ($0 \le x_i, y_i \le 10^9$) — position of the i-th bottle.

It's guaranteed that positions of Adil, Bera, recycling bin and all bottles are distinct.

Output

Print one real number — the minimum possible total distance Adil and Bera need to walk in order to put all bottles into recycling bin. Your answer will be considered correct if its absolute or relative error does not exceed 10^{-6}.

Namely: let's assume that your answer is a, and the answer of the jury is b. The checker program will consider your answer correct if .

Examples

input
3 1 1 2 0 0 3 1 1 2 1 2 3
output
11.084259940083

input
5 0 4 2 2 0 5 5 2 3 0 5 5 3 5 3 3
output
33.121375178000

Note

Consider the first sample.

Adil will use the following path: .

Bera will use the following path: .

Adil's path will be units long, while Bera's path will be units long.

D. Robin Hood

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

We all know the impressive story of Robin Hood. Robin Hood uses his archery skills and his wits to steal the money from rich, and return it to the poor.

There are n citizens in Kekoland, each person has c_i coins. Each day, Robin Hood will take exactly 1 coin from the richest person in the city and he will give it to the poorest person (poorest person right after taking richest's 1 coin). In case the choice is not unique, he will select one among them at random. Sadly, Robin Hood is old and want to retire in k days. He decided to spend these last days with helping poor people.

After taking his money are taken by Robin Hood richest person may become poorest person as well, and it might even happen that Robin Hood will give his money back. For example if all people have same number of coins, then next day they will have same number of coins too.

Your task is to find the difference between richest and poorest persons wealth after k days. Note that the choosing at random among richest and poorest doesn't affect the answer.

Input

The first line of the input contains two integers n and k ($1 \le n \le 500\,000$, $0 \le k \le 10^9$) — the number of citizens in Kekoland and the number of days left till Robin Hood's retirement.

The second line contains n integers, the i-th of them is c_i ($1 \le c_i \le 10^9$) — initial wealth of the i-th person.

Output

Print a single line containing the difference between richest and poorest peoples wealth.

Examples

input
4 1 1 1 4 2
output
2

input
3 1 2 2 2
output
0

Note

Lets look at how wealth changes through day in the first sample.

1. $[1, 1, 4, 2]$
2. $[2, 1, 3, 2]$ or $[1, 2, 3, 2]$

So the answer is $3 - 1 = 2$

In second sample wealth will remain the same for each person.

E. Ultimate Weirdness of an Array

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

Yasin has an array a containing n integers. Yasin is a 5 year old, so he loves ultimate weird things.

Yasin denotes *weirdness* of an array as maximum $gcd(a_i, a_j)$ value among all $1 \le i < j \le n$. For $n \le 1$ weirdness is equal to 0, $gcd(x, y)$ is the greatest common divisor of integers x and y.

He also defines the *ultimate weirdness* of an array. Ultimate weirdness is where $f(i, j)$ is weirdness of the new array a obtained by removing all elements between i and j inclusive, so new array is $[a_1 \ldots a_{i-1}, a_{j+1} \ldots a_n]$.

Since 5 year old boys can't code, Yasin asks for your help to find the value of ultimate weirdness of the given array a!

Input

The first line of the input contains a single integer n $(1 \le n \le 200\,000)$ — the number of elements in a.

The next line contains n integers a_i $(1 \le a_i \le 200\,000)$, where the i-th number is equal to the i-th element of the array a. It is guaranteed that all a_i are distinct.

Output

Print a single line containing the value of ultimate weirdness of the array a.

Example

input
3 2 6 3
output
6

Note

Consider the first sample.

- $f(1, 1)$ is equal to 3.
- $f(2, 2)$ is equal to 1.
- $f(3, 3)$ is equal to 2.
- $f(1, 2), f(1, 3)$ and $f(2, 3)$ are equal to 0.

Thus the answer is $3 + 0 + 0 + 1 + 0 + 2 = 6$.