CODEFORCES β
Sponsored by Telegram

Codeforces Round #276 (Div. 2)

A. Factory

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

One industrial factory is reforming working plan. The director suggested to set a mythical detail production norm. If at the beginning of the day there were x details in the factory storage, then by the end of the day the factory has to produce $x \bmod m$ (remainder after dividing x by m) more details. Unfortunately, no customer has ever bought any mythical detail, so all the details produced stay on the factory.

The board of directors are worried that the production by the given plan may eventually stop (that means that there will be a moment when the current number of details on the factory is divisible by m).

Given the number of details a on the first day and number m check if the production stops at some moment.

Input

The first line contains two integers a and m ($1 \leq a, m \leq 10^5$).

Output

Print "`Yes`" (without quotes) if the production will eventually stop, otherwise print "`No`".

Sample test(s)

input
```
1 5
```
output
```
No
```

input
```
3 6
```
output
```
Yes
```

B. Valuable Resources

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

Many computer strategy games require building cities, recruiting army, conquering tribes, collecting resources. Sometimes it leads to interesting problems.

Let's suppose that your task is to build a square city. The world map uses the Cartesian coordinates. The sides of the city should be parallel to coordinate axes. The map contains mines with valuable resources, located at some points with integer coordinates. The sizes of mines are relatively small, i.e. they can be treated as points. The city should be built in such a way that all the mines are inside or on the border of the city square.

Building a city takes large amount of money depending on the size of the city, so you have to build the city with the minimum area. Given the positions of the mines find the minimum possible area of the city.

Input

The first line of the input contains number n — the number of mines on the map ($2 \le n \le 1000$). Each of the next n lines contains a pair of integers x_i and y_i — the coordinates of the corresponding mine ($-10^9 \le x_i, y_i \le 10^9$). All points are pairwise distinct.

Output

Print the minimum area of the city that can cover all the mines with valuable resources.

Sample test(s)

input

```
2
0 0
2 2
```

output

```
4
```

input

```
2
0 0
0 3
```

output

```
9
```

C. Bits

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

Let's denote as $\text{popcount}(x)$ the number of bits set ('1' bits) in the binary representation of the non-negative integer x.

You are given multiple queries consisting of pairs of integers l and r. For each query, find the x, such that $l \le x \le r$, and $\text{popcount}(x)$ is maximum possible. If there are multiple such numbers find the smallest of them.

Input

The first line contains integer n — the number of queries ($1 \le n \le 10000$).

Each of the following n lines contain two integers l_i, r_i — the arguments for the corresponding query ($0 \le l_i \le r_i \le 10^{18}$).

Output

For each query print the answer in a separate line.

Sample test(s)

input

```
3
1 2
2 4
1 10
```

output

```
1
3
7
```

Note

The binary representations of numbers from 1 to 10 are listed below:

$1_{10} = 1_2$

$2_{10} = 10_2$

$3_{10} = 11_2$

$4_{10} = 100_2$

$5_{10} = 101_2$

$6_{10} = 110_2$

$7_{10} = 111_2$

$8_{10} = 1000_2$

$9_{10} = 1001_2$

$10_{10} = 1010_2$

D. Maximum Value

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

You are given a sequence a consisting of n integers. Find the maximum possible value of $a_i \bmod a_j$ (integer remainder of a_i divided by a_j), where $1 \le i, j \le n$ and $a_i \ge a_j$.

Input

The first line contains integer n — the length of the sequence ($1 \le n \le 2 \cdot 10^5$).

The second line contains n space-separated integers a_i ($1 \le a_i \le 10^6$).

Output

Print the answer to the problem.

Sample test(s)

input

```
3
3 4 5
```

output

```
2
```

E. Strange Sorting

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

How many specific orders do you know? Ascending order, descending order, order of ascending length, order of ascending polar angle... Let's have a look at another specific order: d*-sorting*. This sorting is applied to the strings of length at least d, where d is some positive integer. The characters of the string are sorted in following manner: first come all the 0-th characters of the initial string, then the 1-st ones, then the 2-nd ones and so on, in the end go all the $(d - 1)$-th characters of the initial string. By the i-th characters we mean all the character whose positions are exactly i modulo d. If two characters stand on the positions with the same remainder of integer division by d, their relative order after the sorting shouldn't be changed. The string is zero-indexed. For example, for string '`qwerty`':

Its 1-sorting is the string '`qwerty`' (all characters stand on 0 positions),

Its 2-sorting is the string '`qetwry`' (characters '`q`', '`e`' and '`t`' stand on 0 positions and characters '`w`', '`r`' and '`y`' are on 1 positions),

Its 3-sorting is the string '`qrwtey`' (characters '`q`' and '`r`' stand on 0 positions, characters '`w`' and '`t`' stand on 1 positions and characters '`e`' and '`y`' stand on 2 positions),

Its 4-sorting is the string '`qtwyer`',

Its 5-sorting is the string '`qywert`'.

You are given string S of length n and m *shuffling* operations of this string. Each *shuffling* operation accepts two integer arguments k and d and transforms string S as follows. For each i from 0 to $n - k$ in the increasing order we apply the operation of d-sorting to the substring $S[i..i + k - 1]$. Here $S[a..b]$ represents a substring that consists of characters on positions from a to b inclusive.

After each *shuffling* operation you need to print string S.

Input

The first line of the input contains a non-empty string S of length n, consisting of lowercase and uppercase English letters and digits from 0 to 9.

The second line of the input contains integer m – the number of *shuffling* operations $(1 \le m \cdot n \le 10^6)$.

Following m lines contain the descriptions of the operations consisting of two integers k and d $(1 \le d \le k \le n)$.

Output

After each operation print the current state of string S.

Sample test(s)

input

```
qwerty
3
4 2
6 3
5 2
```

output

```
qertwy
qtewry
qetyrw
```

Note

Here is detailed explanation of the sample. The first modification is executed with arguments $k = 4$, $d = 2$. That means that you need to apply 2-sorting for each substring of length 4 one by one moving from the left to the right. The string will transform in the following manner:

qwerty → **qewr**ty → q**erwt**y → qe**rtwy**

Thus, string S equals '`qertwy`' at the end of first query.

The second modification is executed with arguments $k = 6$, $d = 3$. As a result of this operation the whole string S is replaced by its 3-sorting:

qertwy → **qtewry**

The third modification is executed with arguments $k = 5$, $d = 2$.

qtewry → **qertw**y → q**etyrw**