A. Text Volume

1 second, 256 megabytes

You are given a text of single-space separated words, consisting of small and capital Latin letters.

Volume of the word is number of capital letters in the word. *Volume* of the text is maximum *volume* of all words in the text.

Calculate the *volume* of the given text.

Input

The first line contains one integer number n $(1 \le n \le 200)$ — length of the text.

The second line contains text of single-space separated words $s_1, s_2, ..., s_i$, consisting only of small and capital Latin letters.

Output

Print one integer number — *volume* of text.

input
```
7
NonZERO
```
output
```
5
```

input
```
24
this is zero answer text
```
output
```
0
```

input
```
24
Harbour Space University
```
output
```
1
```

In the first example there is only one word, there are 5 capital letters in it.

In the second example all of the words contain 0 capital letters.

B. Flag of Berland

1 second, 256 megabytes

The flag of Berland is such rectangular field $n \times m$ that satisfies following conditions:

- Flag consists of three colors which correspond to letters `'R'`, `'G'` and `'B'`.
- Flag consists of three equal in width and height stripes, parralel to each other and to sides of the flag. Each stripe has **exactly one color**.
- Each color should be used in **exactly one stripe**.

You are given a field $n \times m$, consisting of characters `'R'`, `'G'` and `'B'`. Output "YES" (without quotes) if this field corresponds to correct flag of Berland. Otherwise, print "NO" (without quotes).

Input

The first line contains two integer numbers n and m $(1 \le n, m \le 100)$ — the sizes of the field.

Each of the following n lines consisting of m characters `'R'`, `'G'` and `'B'` — the description of the field.

Output

Print "YES" (without quotes) if the given field corresponds to correct flag of Berland . Otherwise, print "NO" (without quotes).

input

```
6 5
RRRRR
RRRRR
BBBBB
BBBBB
GGGGG
GGGGG
```

output

```
YES
```

input

```
4 3
BRG
BRG
BRG
BRG
```

output

```
YES
```

input

```
6 7
RRRGGGG
RRRGGGG
RRRGGGG
RRRBBBB
RRRBBBB
RRRBBBB
```

output

```
NO
```

input

```
4 4
RRRR
RRRR
BBBB
GGGG
```

output

```
NO
```

The field in the third example doesn't have three parralel stripes.

Rows of the field in the fourth example are parralel to each other and to borders. But they have different heights — `2`, `1` and `1`.

C. Two Seals

1 second, 256 megabytes

One very important person has a piece of paper in the form of a rectangle $a \times b$.

Also, he has n seals. Each seal leaves an impression on the paper in the form of a rectangle of the size $x_i \times y_i$. Each impression must be parallel to the sides of the piece of paper (but seal can be rotated by 90 degrees).

A very important person wants to choose two different seals and put them two impressions. Each of the selected seals puts exactly one impression. Impressions should not overlap (but they can touch sides), and the total area occupied by them should be the largest possible. What is the largest area that can be occupied by two seals?

Input

The first line contains three integer numbers n, a and b $(1 \leq n, a, b \leq 100)$.

Each of the next n lines contain two numbers x_i, y_i $(1 \leq x_i, y_i \leq 100)$.

Output

Print the largest total area that can be occupied by two seals. If you can not select two seals, print `0`.

input

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

output

```
4
```

input

```
4 10 9
2 3
1 1
5 10
9 11
```

output

```
56
```

input

```
3 10 10
6 6
7 7
20 5
```

output

```
0
```

In the first example you can rotate the second seal by 90 degrees. Then put impression of it right under the impression of the first seal. This will occupy all the piece of paper.

In the second example you can't choose the last seal because it doesn't fit. By choosing the first and the third seals you occupy the largest area.

In the third example there is no such pair of seals that they both can fit on a piece of paper.

D. Round Subset

2 seconds, 256 megabytes

Let's call the *roundness* of the number the number of zeros to which it ends.

You have an array of n numbers. You need to choose a subset of exactly k numbers so that the *roundness* of the product of the selected numbers will be maximum possible.

Input

The first line contains two integer numbers n and k ($1 \le n \le 200, 1 \le k \le n$).

The second line contains n space-separated integer numbers $a_1, a_2, ..., a_n$ ($1 \le a_i \le 10^{18}$).

Output

Print maximal roundness of product of the chosen subset of length k.

input

```
3 2
50 4 20
```

output

```
3
```

input

```
5 3
15 16 3 25 9
```

output

```
3
```

input

```
3 3
9 77 13
```

output

```
0
```

In the first example there are `3` subsets of `2` numbers. $[50, 4]$ has product `200` with *roundness* `2`, $[4, 20]$ — product `80`, *roundness* `1`, $[50, 20]$ — product `1000`, *roundness* `3`.

In the second example subset $[15, 16, 25]$ has product `6000`, *roundness* `3`.

In the third example all subsets has product with *roundness* `0`.

E. Vasya's Function

1 second, 256 megabytes

Vasya is studying number theory. He has denoted a function $f(a, b)$ such that:

- $f(a, 0) = 0$;
- $f(a, b) = 1 + f(a, b - gcd(a, b))$, where $gcd(a, b)$ is the greatest common divisor of a and b.

Vasya has two numbers x and y, and he wants to calculate $f(x, y)$. He tried to do it by himself, but found out that calculating this function the way he wants to do that might take very long time. So he decided to ask you to implement a program that will calculate this function swiftly.

Input

The first line contains two integer numbers x and y ($1 \le x, y \le 10^{12}$).

Output

Print $f(x, y)$.

input
3 5
output
3

input
6 3
output
1

F. Prefix Sums

1 second, 256 megabytes

Consider the function $p(x)$, where x is an array of m integers, which returns an array y consisting of $m + 1$ integers such that y_i is equal to the sum of first i elements of array x ($0 \le i \le m$).

You have an infinite sequence of arrays $A^0, A^1, A^2...$, where A^0 is given in the input, and for each $i \ge 1$ $A^i = p(A^{i-1})$. Also you have a positive integer k. You have to find minimum possible i such that A^i contains a number which is larger or equal than k.

Input

The first line contains two integers n and k ($2 \le n \le 200000$, $1 \le k \le 10^{18}$). n is the size of array A^0.

The second line contains n integers $A^0{}_0, A^0{}_1... A^0{}_{n-1}$ — the elements of A^0 ($0 \le A^0{}_i \le 10^9$). At least two elements of A^0 are positive.

Output

Print the minimum i such that A^i contains a number which is larger or equal than k.

input
2 2 1 1
output
1

input
3 6 1 1 1
output
2

input
3 1 1 0 1
output
0

G. Functions On The Segments

5 seconds, 1024 megabytes

You have an array f of n functions.The function $f_i(x)$ ($1 \le i \le n$) is characterized by parameters: x_1, x_2, y_1, a, b, y_2 and take values:

- y_1, if $x \le x_1$.
- $a \cdot x + b$, if $x_1 < x \le x_2$.
- y_2, if $x > x_2$.

There are m queries. Each query is determined by numbers l, r and x. For a query with number i ($1 \le i \le m$), you need to calculate the sum of all $f_j(x_i)$ where $l \le j \le r$. The value of x_i is calculated as follows: $x_i = (x + last)$ mod 10^9, where $last$ is the answer to the query with number $i - 1$. The value of $last$ equals 0 if $i = 1$.

Input

First line contains one integer number n ($1 \le n \le 75000$).

Each of the next n lines contains six integer numbers: x_1, x_2, y_1, a, b, y_2 ($0 \le x_1 < x_2 \le 2 \cdot 10^5$, $0 \le y_1, y_2 \le 10^9$, $0 \le a, b \le 10^4$).

Next line contains one integer number m ($1 \le m \le 500000$).

Each of the next m lines contains three integer numbers: l, r and x ($1 \le l \le r \le n$, $0 \le x \le 10^9$).

input

```
1
1 2 1 4 5 10
1
1 1 2
```

output

```
13
```

input

```
3
2 5 1 1 1 4
3 6 8 2 5 7
1 3 5 1 4 10
3
1 3 3
2 3 2
1 2 5
```

output

```
19
17
11
```