A. Bear and Raspberry

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

The bear decided to store some raspberry for the winter. He cunningly found out the price for a barrel of honey in kilos of raspberry for each of the following n days. According to the bear's data, on the i-th $(1 \le i \le n)$ day, the price for one barrel of honey is going to is x_i kilos of raspberry.

Unfortunately, the bear has neither a honey barrel, nor the raspberry. At the same time, the bear's got a friend who is ready to lend him a barrel of honey for exactly one day for c kilograms of raspberry. That's why the bear came up with a smart plan. He wants to choose some day d $(1 \le d < n)$, lent a barrel of honey and immediately (on day d) sell it according to a daily exchange rate. The next day $(d + 1)$ the bear wants to buy a new barrel of honey according to a daily exchange rate (as he's got some raspberry left from selling the previous barrel) and immediately (on day $d + 1$) give his friend the borrowed barrel of honey as well as c kilograms of raspberry for renting the barrel.

The bear wants to execute his plan at most once and then hibernate. What maximum number of kilograms of raspberry can he earn? Note that if at some point of the plan the bear runs out of the raspberry, then he won't execute such a plan.

Input

The first line contains two space-separated integers, n and c $(2 \le n \le 100, 0 \le c \le 100)$, — the number of days and the number of kilos of raspberry that the bear should give for borrowing the barrel.

The second line contains n space-separated integers $x_1, x_2, \ldots, x_n$ $(0 \le x_i \le 100)$, the price of a honey barrel on day i.

Output

Print a single integer — the answer to the problem.

Examples

input
```
5 1
5 10 7 3 20
```
output
```
3
```

input
```
6 2
100 1 10 40 10 40
```
output
```
97
```

input
```
3 0
1 2 3
```
output
```
0
```

Note

In the first sample the bear will lend a honey barrel at day 3 and then sell it for 7. Then the bear will buy a barrel for 3 and return it to the friend. So, the profit is (7 - 3 - 1) = 3.

In the second sample bear will lend a honey barrel at day 1 and then sell it for 100. Then the bear buy the barrel for 1 at the day 2. So, the profit is (100 - 1 - 2) = 97.

B. Bear and Strings

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

The bear has a string $s = s_1s_2... s_{|s|}$ (record $|s|$ is the string's length), consisting of lowercase English letters. The bear wants to count the number of such pairs of indices i, j $(1 \le i \le j \le |s|)$, that string $x(i, j) = s_is_{i+1}... s_j$ contains at least one string "bear" as a substring.

String $x(i, j)$ contains string "bear", if there is such index k $(i \le k \le j - 3)$, that $s_k = b$, $s_{k+1} = e$, $s_{k+2} = a$, $s_{k+3} = r$.

Help the bear cope with the given problem.

Input

The first line contains a non-empty string s $(1 \le |s| \le 5000)$. It is guaranteed that the string only consists of lowercase English letters.

Output

Print a single number — the answer to the problem.

Examples

input

```
bearbtear
```

output

```
6
```

input

```
bearaabearc
```

output

```
20
```

Note

In the first sample, the following pairs (i, j) match: $(1, 4), (1, 5), (1, 6), (1, 7), (1, 8), (1, 9)$.

In the second sample, the following pairs (i, j) match:
$(1, 4), (1, 5), (1, 6), (1, 7), (1, 8), (1, 9), (1, 10), (1, 11), (2, 10), (2, 11), (3, 10), (3, 11), (4, 10), (4, 11), (5, 10), (5, 11), (6, 10), (6$
.

C. Bear and Prime Numbers

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Recently, the bear started studying data structures and faced the following problem.

You are given a sequence of integers $x_1, x_2, \dots, x_n$ of length n and m queries, each of them is characterized by two integers l_i, r_i. Let's introduce $f(p)$ to represent the number of such indexes k, that x_k is divisible by p. The answer to the query l_i, r_i is the sum: $\sum_{p \in S(l_i, r_i)} f(p)$, where $S(l_i, r_i)$ is a set of prime numbers from segment $[l_i, r_i]$ (both borders are included in the segment).

Help the bear cope with the problem.

Input

The first line contains integer n $(1 \le n \le 10^6)$. The second line contains n integers $x_1, x_2, \dots, x_n$ $(2 \le x_i \le 10^7)$. The numbers are not necessarily distinct.

The third line contains integer m $(1 \le m \le 50000)$. Each of the following m lines contains a pair of space-separated integers, l_i and r_i $(2 \le l_i \le r_i \le 2 \cdot 10^9)$ — the numbers that characterize the current query.

Output

Print m integers — the answers to the queries on the order the queries appear in the input.

Examples

input

```
6
5 5 7 10 14 15
3
2 11
3 12
4 4
```

output

```
9
7
0
```

input

```
7
2 3 5 7 11 4 8
2
8 10
2 123
```

output

```
0
7
```

Note

Consider the first sample. Overall, the first sample has 3 queries.

1. The first query $l = 2$, $r = 11$ comes. You need to count $f(2) + f(3) + f(5) + f(7) + f(11) = 2 + 1 + 4 + 2 + 0 = 9$.
2. The second query comes $l = 3$, $r = 12$. You need to count $f(3) + f(5) + f(7) + f(11) = 1 + 4 + 2 + 0 = 7$.
3. The third query comes $l = 4$, $r = 4$. As this interval has no prime numbers, then the sum equals 0.

D. Bear and Floodlight

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

One day a bear lived on the Oxy axis. He was afraid of the dark, so he couldn't move at night along the plane points that aren't lit. One day the bear wanted to have a night walk from his house at point $(l, 0)$ to his friend's house at point $(r, 0)$, along the segment of length $(r - l)$. Of course, if he wants to make this walk, he needs each point of the segment to be lit. That's why the bear called his friend (and yes, in the middle of the night) asking for a very delicate favor.

The Oxy axis contains n floodlights. Floodlight i is at point (x_i, y_i) and can light any angle of the plane as large as a_i degree with vertex at point (x_i, y_i). The bear asked his friend to turn the floodlights so that he (the bear) could go as far away from his house as possible during the walking along the segment. His kind friend agreed to fulfill his request. And while he is at it, the bear wonders: what is the furthest he can go away from his house? Hep him and find this distance.

Consider that the plane has no obstacles and no other light sources besides the floodlights. The bear's friend cannot turn the floodlights during the bear's walk. Assume that after all the floodlights are turned in the correct direction, the bear goes for a walk and his friend goes to bed.

Input

The first line contains three space-separated integers n, l, r $(1 \le n \le 20;\ -10^5 \le l \le r \le 10^5)$. The i-th of the next n lines contain three space-separated integers x_i, y_i, a_i $(-1000 \le x_i \le 1000;\ 1 \le y_i \le 1000;\ 1 \le a_i \le 90)$ — the floodlights' description.

Note that two floodlights can be at the same point of the plane.

Output

Print a single real number — the answer to the problem. The answer will be considered correct if its relative or absolute error doesn't exceed 10^{-6}.

Examples

input
```
2 3 5
3 1 45
5 1 45
```
output
```
2.000000000
```

input
```
1 0 1
1 1 30
```
output
```
0.732050808
```

input
```
1 0 1
1 1 45
```
output
```
1.000000000
```

input
```
1 0 2
0 2 90
```
output
```
2.000000000
```

Note

In the first sample, one of the possible solutions is:

In the second sample, a single solution is:

In the third sample, a single solution is:

E. Bear in the Field

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

Our bear's forest has a checkered field. The checkered field is an $n \times n$ table, the rows are numbered from 1 to n from top to bottom, the columns are numbered from 1 to n from left to right. Let's denote a cell of the field on the intersection of row x and column y by record (x, y). Each cell of the field contains growing raspberry, at that, the cell (x, y) of the field contains $x + y$ raspberry bushes.

The bear came out to walk across the field. At the beginning of the walk his speed is (dx, dy). Then the bear spends exactly t seconds on the field. Each second the following takes place:

- Let's suppose that at the current moment the bear is in cell (x, y).
- First the bear eats the raspberry from all the bushes he has in the current cell. After the bear eats the raspberry from k bushes, he increases each component of his speed by k. In other words, if before eating the k bushes of raspberry his speed was (dx, dy), then after eating the berry his speed equals $(dx + k, dy + k)$.
- Let's denote the current speed of the bear (dx, dy) (it was increased after the previous step). Then the bear moves from cell (x, y) to cell $(((x + dx - 1) \bmod n) + 1, ((y + dy - 1) \bmod n) + 1)$.
- Then one additional raspberry bush grows in each cell of the field.

You task is to predict the bear's actions. Find the cell he ends up in if he starts from cell (sx, sy). Assume that each bush has infinitely much raspberry and the bear will never eat all of it.

Input

The first line of the input contains six space-separated integers: n, sx, sy, dx, dy, t $(1 \le n \le 10^9;\ 1 \le sx, sy \le n;\ -100 \le dx, dy \le 100;\ 0 \le t \le 10^{18})$.

Output

Print two integers — the coordinates of the cell the bear will end up in after t seconds.

Examples

input

```
5 1 2 0 1 2
```

output

```
3 1
```

input

```
1 1 1 -1 -1 2
```

output

```
1 1
```

Note

Operation $a \bmod b$ means taking the remainder after dividing a by b. Note that the result of the operation is always non-negative. For example, $(-1) \bmod 3 = 2$.

In the first sample before the first move the speed vector will equal (3,4) and the bear will get to cell (4,1). Before the second move the speed vector will equal (9,10) and he bear will get to cell (3,1). Don't forget that at the second move, the number of berry bushes increased by 1.

In the second sample before the first move the speed vector will equal (1,1) and the bear will get to cell (1,1). Before the second move, the speed vector will equal (4,4) and the bear will get to cell (1,1). Don't forget that at the second move, the number of berry bushes increased by 1.