Contest 3 : HN

A. University Career Fair

1 second, 256 megabytes

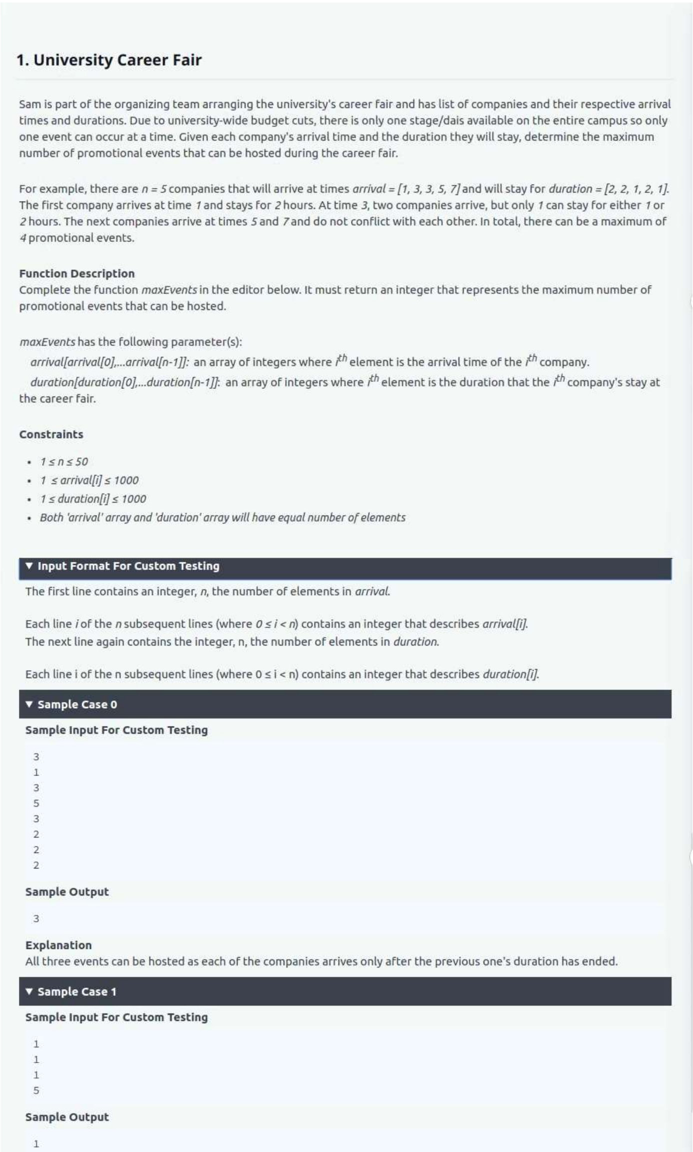
Problem Description

Sam is part of the organizing team arranging the university's career fair and has a list of companies and their respective arrival times and durations. Due to university-wide budget cuts, there is only one stage/dais available on the entire campus so only one event can occur at a time. Given each company's arrival time and the duration for which they will stay, determine the maximum number of promotional events that can be hosted during the career fair.

For example, there are $n = 5$ companies that will arrive at times $arrival = [1, 3, 3, 5, 7]$ and will stay for $duration = [2, 2, 1, 2, 1]$. The first company arrives at a time of 1 and stays for 2 hours. At the time 3, two companies arrive, but only 1 can stay for either 1 or 2 hours. The next companies arrive at times 5 and 7 and do not conflict with each other. In total, there can be a maximum of 4 promotional event.

Constraints

- $1 \leq n \leq 5 * 10^4$
- $1 \leq arrival[i] \leq 1000$
- $1 \leq duration[i] \leq 1000$
- Both $arrival$ array and $duration$ array will have equal number of elements .



1. University Career Fair

Sam is part of the organizing team arranging the university's career fair and has list of companies and their respective arrival times and durations. Due to university-wide budget cuts, there is only one stage/dais available on the entire campus so only one event can occur at a time. Given each company's arrival time and the duration they will stay, determine the maximum number of promotional events that can be hosted during the career fair.

For example, there are *n = 5* companies that will arrive at times *arrival = [1, 3, 3, 5, 7]* and will stay for *duration = [2, 2, 1, 2, 1]*. The first company arrives at time *1* and stays for *2* hours. At time *3*, two companies arrive, but only *1* can stay for either *1* or *2* hours. The next companies arrive at times *5* and *7* and do not conflict with each other. In total, there can be a maximum of *4* promotional events.

Function Description

Complete the function *maxEvents* in the editor below. It must return an integer that represents the maximum number of promotional events that can be hosted.

maxEvents has the following parameter(s):

arrival[arrival[0],...arrival[n-1]]: an array of integers where i^{th} element is the arrival time of the i^{th} company.

duration[duration[0],...duration[n-1]]: an array of integers where i^{th} element is the duration that the i^{th} company's stay at the career fair.

Constraints

- *1 ≤ n ≤ 50*
- *1 ≤ arrival[i] ≤ 1000*
- *1 ≤ duration[i] ≤ 1000*
- *Both 'arrival' array and 'duration' array will have equal number of elements*

▼ Input Format For Custom Testing

The first line contains an integer, *n*, the number of elements in *arrival*.

Each line *i* of the *n* subsequent lines (where *0 ≤ i < n*) contains an integer that describes *arrival[i]*.
The next line again contains the integer, n, the number of elements in *duration*.

Each line i of the n subsequent lines (where 0 ≤ i < n) contains an integer that describes *duration[i]*.

▼ Sample Case 0

Sample Input For Custom Testing

```
3
1
3
5
3
2
2
2
```

Sample Output

```
3
```

Explanation

All three events can be hosted as each of the companies arrives only after the previous one's duration has ended.

▼ Sample Case 1

Sample Input For Custom Testing

```
1
1
1
5
```

Sample Output

```
1
```

Input

The first line contains an integer n, denoting the number of elements in $arrival$.

Each line i of the n subsequent lines (where $0 \leq i < n$) contains an integer that describes $arrival[i]$.

The next line again contains the integer n, denoting the number of elements in $duration$.

Each line i of the n subsequent lines (where $0 \leq i < n$) contains an integer that describes $duration[i]$.

Output

Print the maximum number of promotional events that can be hosted.

input
```
3
1
3
5
3
2
2
2
```
output
```
3
```

input
```
1
1
1
5
```
output
```
1
```

For the first case, all three events can be hosted as each of the companies arrive only after the previous one's duration has ended.

For the second case, only one company is present and its event can be hosted with no conflict.

B. Paint the Ceiling

1 second, 256 megabytes

Problem Description

You want to build yourself a house. The building company you hired can only build houses with sides from their specific set s. That means they can build you a square house or a rectangular one but if and only if its length and width belong to the set s.

This month, they have a special promotion: they will paint the ceiling of a new house for free...but only if its area is not more than a. You want them to do it for free but you also want to be sure that the house will be comfortable and not too small. How many possible house configurations can you create to have the ceiling painted for free given the side lengths offered?

There is a method to how the company decided what lengths of sides to produce. To determine n lengths of wall segments to offer, they start with a seed value s_0, some variables k, b and m, and use the following equation to determine all other side lengths $s[i]$:

$$s[i] = (((k * s[i-1] + b)\%m) + 1 + s[i-1]) \qquad \forall 1 \leq i < n$$

For example, you are given $s[0] = s_0 = 2$ and they will produce $n = 3$ total wall lengths. If $k = 3, b = 3$ and $m = 2$ we have :

$[s,\ calculation, result]$

Iteration 1 : $[[2],\ ((3*2+3)\%2)+1+2,\ 4]$

Iteration 2 : $[[2,4],\ ((3*4+3)\%2)+1+4,\ 6]$

Finally, we have $[2,\ 4,\ 6]$

Now that we have our set of lengths, we can brute force the solution using the following tests assuming $a = 15$:

$s = [2,\ 4,\ 6]$

$[s_1,\ \ s_2,\ \ s_1 * s_2,\ \ s_1 * s_2 \le a]$

$[2,\ 2,\ 4,\ True]$

$[2,\ 4,\ 8,\ True]$

$[2,\ 6,\ 12,\ True]$

$[4,\ 2,\ 8,\ True]$

$[4,\ 4,\ 16,\ False]$

$[4,\ 6,\ 24,\ False]$

$[6,\ 2,\ 12,\ True]$

$[6,\ 4,\ 24,\ False]$

$[6,\ 6,\ 36,\ False]$

There are 5 combinations that will result in a free paint job.

Constraints

- $1 \le n \le 2 * 10^7$
- $1 \le s_0 \le 10^3$
- $1 \le k,\ b,\ m \le 10^9$
- $1 \le a \le 10^{18}$

12. Paint the ceiling

You want to build yourself a house. The building company you hired can only build houses with sides from their specific set *s*. That means they can build you a square house or a rectangular one but if and only if its length and width belong to the set *s*.

This month, they have a special promotion: they will paint the ceiling of a new house for free...but only if its area is not more than *a*. You want them to do it for free but you also want to be sure that the house will be comfortable and not too small. How many possible house configurations can you create to have the ceiling painted for free given the side lengths offered?

There is a method to how the company decides what lengths of sides to produce. To determine *n* lengths of wall segments to offer, they start with a seed value *s0*, some variables *k*, *b* and *m*, and use the following equation to determine all other side lengths *s[i]*:

*s[i]=((k*s[i-1]+b) mod m)+1+s[i-1])* for $1 \le i < n$

For example, you are given *s[0] = s0 = 2* and they will produce *n = 3* total wall lengths. If *k =*

*s[i]=((k*s[i-1]+b) mod m)+1+s[i-1])* for $1 \le i < n$

For example, you are given *s[0] = s0 = 2* and they will produce *n = 3* total wall lengths. If *k =* *3, b = 3* and *m = 2* we have:

```
s       i       calculation     resu
[2]     1       ((3*2+3)%2)+1+2 4
[2,4]   2       ((3*4+3)%2)+1+4 6
[2,4,6]
```

Now that we have our set of lengths, we can brute force the solution using the following tests assuming *a = 15*:

```
s = [2, 4, 6]

s1      s2      s1*s2   s1*s2 <= a
2       2       4       True
2       4       8       True
2       6       12      True
4       2       8       True
4       4       16      False
4       6       24      False
6       2       12      True
6       4       24      False
6       6       36      False
```

There are *5* combinations that will result in a free paint job. Brute force will not meet the time

Function Description

Complete the function *variantsCount* in the editor below. The function must return an integer that denotes the number of variants that allow you to use the promotion.

variantsCount has the following parameter(s):

n: an integer, the number of wall lengths offered

s0: an integer, the length of the shortest wall

k, b, m: three arbitrary integers

a: a long integer, the largest area that will be painted for free

Constraints

- $1 \le n \le 6 * 10^7$
- $1 \le s[i] \le 10^9,\ 0 \le i < n$
- $1 \le k, b, m \le 10^9$
- $1 \le a \le 10^{18}$

▶ **Input Format for Custom Testing**

▼ **Sample Case 0**

Sample Input

```
3
```

- $1 \le a \le 10^{18}$

▶ **Input Format for Custom Testing**

▼ **Sample Case 0**

Sample Input

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

Sample Output

```
6
```

Explanation

n = 3, s[0] = s0 = 1, k = 1, b = 1, m = 2 and *a = 4*. That means that *s[1]=((1*1+1) mod 2)+1+1 = 2, s[2]=((1*2+1) mod 2) +1+2 = 4.* That yields the following variants: 1*1 (area=1<=4, good); 1*2 (area=2<=4, good); 1*4 (area=4<=, good); 2*1 (area=2<=4, good); 2*2 (area=4<=4, good); 2*4 (area=8>; bad); 4*1 (area=4<=4, good); 4*2 (area=8>4, bad) and 4*4 (area=16>4, bad). 6 of the variants are good and 3 are bad.

Input

The first line contains n denoting the number of wall lengths offered.

The second line contains an integer s_0 denoting the length of the shortest wall .

The next three lines contains integer k, b, m denoting three arbitrary integers .

The last line contains a long integer a denoting the largest area that will be painted for free .

Output

Print the number of variants that allow you to use the promotion .

input
```
3
1
1
1
2
4
```
output
```
6
```

input
```
3
1
2
1
2
4
```
output
```
3
```

For the **first** testcase,

$n = 3,\ s[0] = s_0 = 1,\ k = 1,\ b = 1,\ m = 2$ and $a = 4$.

That means that

$s[1] = ((1 * 1 + 1)\%2) + 1 + 1 = 2$

$s[2] = ((1 * 2 + 1)\%2) + 1 + 2 = 4.$

That yields the following variants.

$1 * 1, (area = 1 \le 4)$ is good.

$1 * 2, (area = 2 \le 4)$ is good.

$1 * 4, (area = 4 \le 4)$ is good.

$2 * 1, (area = 2 \le 4)$ is good.

$2 * 2, (area = 4 \le 4)$ is good.

$2 * 4, (area = 8 > 4)$ is bad.

$4 * 1, (area = 4 \le 4)$ is good.

$4 * 2, (area = 8 > 4)$ is bad.

$4 * 4, (area = 16 > 4)$ is bad.

6 of the variants are good and 3 are bad.

C. University Career Fair (Hard)

1 second, 256 megabytes

Problem Description

Sam is part of the organizing team arranging the university's career fair and has a list of companies and their respective arrival times, durations and the associated $profit$ to the college. Due to university-wide budget cuts, there is only one stage/dais available on the entire campus so only one event can occur at a time. Given this information, determine the maximum amount of profit that the college can earn by hosting the promotional events.

For example, there are $n = 5$ companies that will arrive at times $arrival = [1, 3, 3, 5, 7]$ and will stay for $duration = [2, 2, 1, 2, 1]$, The profit associated is $profit = [2, 3, 4, 2, 2]$. The first company arrives at a time of 1 and stays for 2 hours. At the time 3, two companies arrive, but only 1 can stay for either 1 or 2 hours. Of course, we'd like to host company 3, as it gives us a better profit. The next companies arrive at times 5 and 7 and do not conflict with each other. In total, there can be a maximum of 4 promotional event, with the total profit as $(2 + 4 + 2 + 2) = 10$

Constraints

- $1 \le n \le 5 * 10^4$
- $1 \le arrival[i] \le 10^9$
- $1 \le duration[i] \le 10^9$
- $2 \le arrival[i] + duration[i] \le 10^9$
- $1 \le profit[i] \le 10^4$

1. University Career Fair

Sam is part of the organizing team arranging the university's career fair and has list of companies and their respective arrival times and durations. Due to university-wide budget cuts, there is only one stage/dais available on the entire campus so only one event can occur at a time. Given each company's arrival time and the duration they will stay, determine the maximum number of promotional events that can be hosted during the career fair.

For example, there are *n = 5* companies that will arrive at times *arrival = [1, 3, 3, 5, 7]* and will stay for *duration = [2, 2, 1, 2, 1]*. The first company arrives at time *1* and stays for *2* hours. At time *3*, two companies arrive, but only *1* can stay for either *1* or *2* hours. The next companies arrive at times *5* and *7* and do not conflict with each other. In total, there can be a maximum of *4* promotional events.

Function Description

Complete the function *maxEvents* in the editor below. It must return an integer that represents the maximum number of promotional events that can be hosted.

maxEvents has the following parameter(s):

arrival[arrival[0],...arrival[n-1]]: an array of integers where i^{th} element is the arrival time of the i^{th} company.

duration[duration[0],...duration[n-1]]: an array of integers where i^{th} element is the duration that the i^{th} company's stay at the career fair.

Constraints

- *1 ≤ n ≤ 50*
- *1 ≤ arrival[i] ≤ 1000*
- *1 ≤ duration[i] ≤ 1000*
- *Both 'arrival' array and 'duration' array will have equal number of elements*

Input

The first line contains an integer n, denoting the number of elements in $arrival$.

Each line i of the n subsequent lines (where $0 \leq i < n$) contains an integer that describes $arrival[i]$.

The next line again contains the integer n, denoting the number of elements in $duration$.

Each line i of the n subsequent lines (where $0 \leq i < n$) contains an integer that describes $duration[i]$.

The next line again contains the integer n, denoting the number of elements in $profit$.

Each line i of the n subsequent lines (where $0 \leq i < n$) contains an integer that describes $profit[i]$.

Output

Print the maximum profit that you can earn while hosting the promotional events in your college.

input

```
5
1
3
3
5
7
5
2
2
1
2
1
5
2
3
4
2
2
```

output

```
10
```

D. Prime Factor Visitation

1 second, 256 megabytes

Problem Description

Constraints

- The first line contains n. The next n lines describe the array **states**.
- The next line contains m. The next m lines describe the array **numbers**.

input

```
10
0
1
1
0
1
1
0
1
1
1
3
3
8
6
```

output

```
0
1
1
0
1
1
0
1
1
1
```

input

```
6
0
0
0
0
0
0
4
2
4
2
6
```

output

```
0
0
1
0
0
1
```

E. Shopper's Delight

1 second, 256 megabytes

Problem Description

input

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

output

```
3
```