Bus Stops

Problem ID: bus

Problem Statement

A school bus travels along a straight road with N bus stops, numbered from 1 to N. At each stop:

- First, some students get off the bus.
- Then, some students get on the bus.

The bus starts empty before stop 1. Your task is to determine the **maximum number of students on the bus at any time**.

In some subtasks, additional rules apply (such as bus capacity). Read the constraints carefully.

Input

- The first line contains an integer N ($1 \leq N \leq 1000$), the number of bus stops.
- The second line contains N integers $\text{on}_1, \text{on}_2, \ldots, \text{on}_N$ ($0 \leq \text{on}_i \leq 10^4$), where on_i is the number of students boarding at stop i.
- The third line contains N integers $\text{off}_1, \text{off}_2, \ldots, \text{off}_N$ ($0 \leq \text{off}_i \leq 10^4$), where off_i is the number of students getting off at stop i.
- The fourth line contains an integer C, the bus capacity. For subtasks where no capacity restriction applies, C will be very large (e.g. 10^9), so it does not affect the result.

Output

Print a single integer: the maximum number of students on the bus at any point in time. In subtasks requiring the stop index, print two integers: `max_students stop_index`.

Subtasks

Subtask	Constraints	Points
1	$N \leq 3$, $\text{on}_i, \text{off}_i \leq 10$, $C = 10^9$	10
2	$N \leq 100$, $\text{on}_i, \text{off}_i \leq 100$, $C = 10^9$	20
3	$N \leq 1000$, $\text{on}_i, \text{off}_i \leq 10^4$, $C = 10^9$	20
4	Same as Subtask 3, but $C \leq 10^4$. If more students attempt to board than seats available, only as many as possible get on. The rest are left behind permanently.	30
5	Same as Subtask 3. Additionally, print the **first stop index** at which the maximum occupancy occurs. Output format: two integers, `max_students stop_index`.	20

Sample Input 1

```
5
0 3 4 0 2
0 0 2 3 4
1000000000
```

Sample Output 1

```
5
```

Explanation for Sample 1

- Stop 1: 0 off, 0 on $\rightarrow$ 0
- Stop 2: 0 off, 3 on $\rightarrow$ 3
- Stop 3: 2 off, 4 on $\rightarrow$ 5
- Stop 4: 3 off, 0 on $\rightarrow$ 2
- Stop 5: 4 off, 2 on $\rightarrow$ 0

Maximum = 5.

Sample Input 2 (capacity example)

```
4
5 5 5 5
0 0 0 0
8
```

Sample Output 2

```
8
```

Explanation for Sample 2

- Stop 1: 5 board → 5
- Stop 2: 5 try to board, but capacity is 8 → 3 board, 2 left behind → total = 8
- Stops 3 and 4: bus remains full at 8

Maximum = 8.

Sample Input 3 (stop index example)

```
6
2 4 0 2 0 0
0 0 2 0 1 4
1000000000
```

Sample Output 3

```
6 2
```

Explanation for Sample 3

- Stop 1: 2 board → 2
- Stop 2: 4 board → 6
- Stop 3: 2 off → 4
- Stop 4: 2 on → 6
- Stop 5: 1 off → 5
- Stop 6: 4 off → 1

Maximum = 6 at stop 2, but maximum **after** that is 6 at stop 4. Since the first maximum is at stop 2, output is `6 2`.

End of Problem 2