

Ada and Salesman

Ada the Ladybug lives in Bugladesh. It is a very specific country - there are some cities, but since the government doesn't "waste" money, they all lie on one simple path.

Ada is working as Traveling Salesman. She travels between cities, buying and selling products. A product has some varying price in each city (same for buy/sale - described later). Ada travels with bike (to avoid payments for travels) so she can carry at most K items at time in her backpack.

She is currently in the first city, and she wants to linearly go to last city. She wants to buy/sell items in a way to maximize her profit. Can you help her?

The system is following: Ada has a bag which can carry at most K items. She travels cities linearly (from city c to city $c+1$). So she is in a city c , she can either immediately move to next city or buy a licence to trade for L_c money. For each city it is also given magical constant P_c . The buying/selling system is pretty weird. Buying an item in city c while actually having $i-1$ items in backpack costs $P_c * i * c \% MOD$. Similarly selling of an item in city c while actually having i items in backpack is for $P_c * i * c \% MOD$. You can buy any number of items in a city and you can also sell any number of items in city. Anyway note that the number of items can't exceed K (and obviously can't be negative).

Input

The first line of input will contain three integers $1 \leq N, K \leq 3000$ and $1 \leq MOD \leq 10^4$.

The next line will contain N integers $0 < L_i \leq 1000$, the cost of licence for each city.

The next line will contain N integers $0 < P_i \leq 1000$, the magical constant for each city.

Cities are numbered from 1.

Output

Output the maximal number of money Ada can gain.

Example Input

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

Example Output

```
0
```

Example Input

```
5 5 6
1 1 1 1 1
1 1 1 1 1
```

Example Output

9

Example Input

```
6 10 11
1 2 3 3 2 1
1 3 2 5 1 7
```

Example Output

19

Example Input

```
9 2 20
6 6 6 6 6 6 6 6
2 4 6 8 2 4 6 8 5
```

Example Output

16

Example Input

```
10 2 20
5 9 3 4 7 5 2 4 7 2
2 1 4 5 4 5 6 3 2 5
```

Example Output

25