保齡球 - Bowling

(10分)

問題敘述

蛋餅是一位優秀的保齡球選手，普通的規則已經無法展現他的強大之處了，因此他發明了屬於自己的保齡球規則。在一般的保齡球規則中，如果全倒就會獲得 10 分 + 下一球的得分 + 下下球的得分，蛋餅將它修改成「如果全倒，就會獲得『上一球的得分乘以 c_1』+『上上球的得分乘以 c_2』$+\ldots+$ K 個球之前的得分乘以 c_K」（注意並沒有基本分 10 分）。你趕到保齡球場的時候，蛋餅已經打完前 K 球了，分數分別是 $a_1, a_2, \ldots, a_K$，並且他很有自信的覺得接下來的每一球都能打出全倒，因為你已經跟他約了吃拉麵，所以蛋餅只會打 N 球。

另外，由於計分板在設計的時候並沒有想到有人這麼會打保齡球，因此他只會顯示蛋餅的總分除以 10^9+9 的餘數。

幫蛋餅計算他打完 N 球之後，計分板上顯示的分數吧。

輸入格式

輸入首行有兩個正整數 N, K，代表蛋餅總共要打的球數以及全倒可以獲得幾球之前的分數，同時也是蛋餅已經打完的球數。

接下來一行有 K 個正整數 $a_1, \ldots, a_K$，代表蛋餅第 i 球打出的分數為 a_i。

接下來一行有 K 個正整數 $c_1, \ldots, c_K$，代表打出全倒時可以獲得前 i 個球的分數乘以 c_i。

輸出格式

輸出一個非負整數於一行，代表蛋餅打完 N 球之後，計分板顯示的分數。

資料範圍

- $1 \le K \le N \le 10^3$
- $1 \le a_i, c_i \le 10^9$

資料範例

輸入範例 1

```
5 3
9 8 3
9 2 5
```

輸出範例 1

```
946
```

輸入範例 2

```
3 1
5
3
```

輸出範例 2

```
65
```

輸入範例 3

```
2 2
1000000000 1000000000
1000000000 1000000000
```

輸出範例 3

```
999999991
```

範例說明

在第一筆範例中，蛋餅第 4 球拿到 $3 \times 9 + 8 \times 2 + 9 \times 5 = 88$ 分，第 5 球拿到 $88 \times 9 + 3 \times 2 + 8 \times 5 = 838$ 分，總分是 $9 + 8 + 3 + 88 + 838 = 946$ 分。

在第二筆範例中，蛋餅第 2 球拿到 $5 \times 3 = 15$ 分，第 3 球拿到 $15 \times 3 = 45$ 分，總分是 $5 + 15 + 45 = 65$ 分。

在第三筆範例中，總分 $10^9 + 10^9 = 2 \times 10^9$ 分，但是計分板只會顯示總分除以 $10^9 + 9$ 的餘數，因此會顯示 999999991 分。

Bowling

(10 points)

Description

Omelet is an excellent bowling player so the ordinary rules can no longer show his strength. Therefore, he invented his own bowling rules. In the general bowling rules, you get "$10+$ the score of te next ball + the score of the next of next ball" if you get a strike, and Omelet modified it to "the score of the previous ball mutiplied by c_1 + the score of the previous of the previous ball multiplied by c_2 + . . . + the score of the ball K before multiplied by c_K"(note that there is no basic score 10). When you arrive at the bowling alley, Omelet has played the first K ball(s), and the score is $a_1, a_2, \ldots, a_K$. He is confident that he can get strike in all the next ball. Because Omelet is going to eat ramen with you, he will only play N ball(s).

Also, since the scoreboard is not designed for someone who can bowl so well, it will only display the reminder of Omelet's total score divided by $10^9 + 9$.

Help Omelet compute the score displayed by the scoreboard after he plays N ball(s).

Input Format

The first line of the input contains two positive integers N, K, indicating the number of ball(s) Omelet is going to play and the number of balls which a strike can get score from. The latter is also the number of ball(s) Omelet has played.

The second line of the input contains K positive integers $a_1, a_2, \ldots, a_K$, indicating the score of the i-th ball Omelet played.

The third line of the input contains K positive integers $c_1, c_2, \ldots, c_K$, indicating the score of the i-th previous ball is multiplied by c_i if Omelet gets a strike.

Output Formet

Output a non-negative integer in one line, indicating that the remainder of the number of the score displayed by the scoreboard after Omelet play N ball(s) divded by $10^9 + 9$.

Constraints

- $1 \leq K \leq N \leq 10^3$
- $1 \leq a_i, c_i \leq 10^9$

Sample Test Cases

Input Example 1

```
5 3
9 8 3
9 2 5
```

Output Example 1

```
946
```

Input Example 2

```
3 1
5
3
```

Output Example 2

```
65
```

Input Example 3

```
2 2
1000000000 1000000000
1000000000 1000000000
```

Output Example 3

```
999999991
```

Example Explanation

In the first example, Omelet gets $3 \times 9 + 8 \times 2 + 9 \times 5 = 88$ points in the 4th ball, and $88 \times 9 + 3 \times 2 + 8 \times 5 = 838$ points in the 5th ball. The total score is $9 + 8 + 3 + 88 + 838 = 946$ points.

In the second example, Omelet gets $5 \times 3 = 15$ points in the 2nd ball, and $15 \times 3 = 45$ points in the 3rd ball. The total score is $5 + 15 + 45 = 65$ points.

In the third example, the total score is$10^9 + 10^9 = 2 \times 10^9$ points, but the scoreboard only displays the remainder of the total score divided by $10^9 + 9$, so the result is 999999991 points.