

# Газонокосилка

Після своїх пригод у фортеці Поенарі Влад повертається додому і, як справжній румун, його перша думка — потрібно нагодувати коня. Кінь не надто вибагливий у їжі, тому Влад використовує свій газон, як основне джерело корму.

Для цього Влад має газонокосарку з баком ємністю  $c$ . Він вирішив розділити свій газон на  $n$  смуг, пронумерованих від 0 до  $n - 1$ , які потрібно косити саме в цьому порядку. Кожна смуга  $i$  містить  $v[i]$  одиниць нескошеної трави, і з певних незрозумілих причин Владу потрібно  $a[i]$  секунд, щоб пройти цю смугу з косаркою.

Після обробки кількох смуг бак косарки може наповнитися. У такому випадку вона припиняє косити траву, залишаючи частину нескошеною. Щоразу, коли бак переповнюється, його потрібно спорожнити, що займає  $b$  секунд, і зробити це можна тільки в кінці смуги. Якщо бак заповнився під час проходження смуги  $i$ , Влад повинен дійти до кінця смуги, спорожнити бак, а потім пройти її ще раз (або стільки разів, скільки потрібно), щоб скосити залишену траву. Наприклад, якщо для смуги  $i$  потрібно пройти по ній 3 рази, щоб позбутися всієї трави, це займе  $a[i] + b + a[i] + b + a[i]$  секунд. **Після того, як буде скошено весь газон, бак обов'язково також потрібно спорожнити.**

Після довгих роздумів і скарг на те, що косіння займе надто багато часу, Влад дійшов висновку, що іноді може бути вигідніше спорожнити бак ще до того, як він повністю заповниться, але він не впевнений, яка стратегія буде найкращою. Тож він звертається до тебе по допомогу.

Враховуючи кількість трави на кожній смузі та кількість секунд, необхідну для проходження кожної смуги, ємність бака і час, потрібний для його спорожнення, — знайди найкращу стратегію, яка дозволить Владу скосити весь газон за мінімальний час.

## Деталі реалізації

Потрібно реалізувати наступну процедуру.

```
long long mow(int n, int c, std::vector<int> &a, std::vector<int> &v);
```

- $n$ : кількість смуг на газоні
- $c$ : загальна ємність бака для збору трави
- $b$ : кількість секунд, необхідна для спорожнення бака
- $a$ : вектор довжини  $n$ , що описує час проходження кожної смуги
- $v$ : вектор довжини  $n$ , що задає кількість трави на кожній смузі

- Ця процедура має повернути одне ціле число — мінімальний час, необхідний для скошування всього газону.
- Процедура викликається рівно один раз для кожного тестового випадку.

## Обмеження

- $1 \leq n \leq 200\,000$
- $1 \leq a[i] \leq 10^9$  (для кожного  $i$  для якого  $0 \leq i < n$ )
- $1 \leq v[i] \leq 10^9$  (для кожного  $i$  для якого  $0 \leq i < n$ )
- $1 \leq b \leq 10^9$
- $1 \leq c \leq 10^9$
- Гарантується, що правильний результат не перевищуватиме  $10^{18}$

## Підзадачі

1. (9 points) Усі задані значення ( $n, b, c, a[i]$  та  $v[i]$ ) не перевищуватимуть 200
2. (16 points)  $n, c \leq 5000$  та  $v[i] \leq 5000$  для всіх  $0 \leq i < n$
3. (36 points)  $c \leq 200\,000$
4. (17 points)  $a[0] = a[1] = \dots = a[n-1]$
5. (22 points) Без додаткових обмежень.

## Приклади

### Приклад 1

Розглянемо наступний виклик:

```
mow(3, 5, 2, {2, 10, 3}, {2, 4, 6})
```

У цьому прикладі є 3 смуги, бак-збирач має місткість 5, а його спустошення займає 2 секунди.

Для цього прикладу Влад покосить першу смугу за 2 секунди. Кількість трави в косарці буде 2. Потім він спустошить косарку за 2 секунди. На першу смугу він витратить 4 секунди.

Потім він пройде по другій смузі. Він покосить 4 одиниці трави. Він вирішив не спустошувати косарку після завершення другої смуги. Час, витрачений на другу смугу — 10 секунд.

На третій смузі він починає косити. Після однієї одиниці трави його косарка заповниться, тож йому доведеться пройти до кінця смуги, спустошити косарку, а потім знову почати косити третю смугу. Зверніть увагу, що після завершення косіння всього газону косарку потрібно буде спорожнити. Час, витрачений на третю смугу:  $3 + 2 + 3 + 2 = 10$  секунд.

Усього він витратить  $4 + 10 + 10 = 24$  секунди. Можна довести, що це оптимальна стратегія, яку Влад використовує для косіння газону.

## Приклад 2

Розглянемо наступний виклик:

```
mow(4, 10, 4, {1, 2, 1, 4}, {3, 2, 6, 7})
```

У цьому прикладі є 4 смуги, бак-збирач має місткість 10, і його спустошення займає 4 секунди.

Оптимальна стратегія — просто пройти перші 3 смуги, після чого бак буде заповнений, і вектор кількостей трави виглядатиме як  $[0, 0, 1, 7]$ . Після цього бак потрібно спустошити, і залишкові 2 смуги буде скошено, а в кінці бак знову спустошується.

Загальний час в секундах цієї стратегії становить:  $a[0] + a[1] + a[2] + b + a[2] + a[3] + b = 17$ .

## Приклад градера

Зчитує введення у такому форматі:

- Рядок 1:  $n$   $c$   $b$
- Рядок 2:  $a[0]$   $a[1]$   $\dots$   $a[n-1]$
- Рядок 3:  $v[0]$   $v[1]$   $\dots$   $v[n-1]$

а виводить результат виклику `mow` з відповідними параметрами.