

# Косачка

След приключенията му в Пойенарската крепост, Влад се прибира вкъщи, и като един истински румънец първата му мисъл е, че трябва да нахрани коня си. Конят не е особено капризен, така че Влад използва тревата от ливадата си като основен източник на храна.

За тази задача Влад има косачка с капацитет  $c$ . Ливадата на Влад е разделена на  $n$  ленти, номерирани с числата от 0 до  $n - 1$ , които той трябва да окоси в точно този ред. Всяка лента  $i$  съдържа количество неокосена трева  $v[i]$ , и по неизвестни причини отнема  $a[i]$  секунди за Влад да мине с косачката си през цялата лента.

След като мине през няколко ленти, косачката може да достигне капацитета си  $c$ , в който случай тя спира да коси, потенциално оставайки част от тревата на тази лента неокосена. Всеки път когато това се случи, резервоарът ѝ трябва да се изпразни, което отнема  $b$  секунди и може да се случи само в края на лента. Ако резервоарът се напълни, докато Влад минава през лента  $i$ , то той трябва да продължи да бута косачката до края на лентата, да изпразни резервоара, и да мине през цялата лента още веднъж (или още няколко пъти, ако отново напълни резервоара), за да довърши останалата неокосена трева. Например, ако за окосяването на лента  $i$  Влад трябва да направи 3 преминавания, то това ще му отнеме  $a[i] + b + a[i] + b + a[i]$  секунди. **След окосяване на цялата ливада, резервоарът също трябва да бъде изпразнен, което отново отнема  $b$  секунди.**

След много размисли и оплаквания за това колко дълго ще отнеме цялата работа, Влад осъзнава, че понякога е по-бързо да изпразни резервоара, дори преди той да се напълни до капацитета си, но за жалост не е сигурен каква е най-добрата стратегия. За това той се нуждае от Вашата помощ.

Дадено Ви е количеството неокосена трева на всяка лента; броят секунди, които отнема да се мине през всяка лента; капацитета на резервоара и времето, което отнема да се изпразни. Намерете най-краткото време, за което Влад може да окоси цялата ливада и да приключи с празна косачка.

## Детайли по имплементацията

Трябва да имплементирате следната функция.

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

- $n$ : броят ленти в ливадата.
- $c$ : капацитета на резервоара на косачката.
- $b$ : броят секунди нужни за изпразване на резервоара (независимо дали е пълен).
- $a$ : вектор с дължина  $n$ , описващ времето нужно за минаване по всяка лента.
- $v$ : вектор с дължина  $n$ , съдържащ количеството неокосена трева на всяка лента.
- Функцията трябва да връща едно число равно на минималното време нужно за окосяване на цялата ливада.
- Функцията ще се извика точно веднъж за всеки тест.

## Ограничения

- $1 \leq n \leq 200\,000$
- $1 \leq a_i \leq 10^9$  (за всяко  $i$ , такова че  $0 \leq i < n$ )
- $1 \leq v_i \leq 10^9$  (за всяко  $i$ , такова че  $0 \leq i < n$ )
- $1 \leq b \leq 10^9$
- $1 \leq c \leq 10^9$
- Гарантирано е, че верният отговор ще бъде най-много  $10^{18}$ .

## Подзадачи

1. (9 точки) Всички стойности ( $n, b, c, a[i]$  и  $v[i]$ ) ще бъдат най-много 200.
2. (16 точки)  $n, c \leq 5000$  и  $v_i \leq 5000$  за всяко  $0 \leq i < n$
3. (36 точки)  $c \leq 200\,000$
4. (17 точки)  $a[0] = a[1] = \dots = a[n-1]$
5. (22 точки) Няма допълнителни ограничения.

## Примери

### Пример 1

Да разгледаме следното извикване:

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

В този пример имаме 3 ленти, резервоарът е с капацитет 5, и отнема 2 секунди да бъде изпразнен. Нека разгледаме едно оптимално решение.

Влад ще окоси първата лента за 2 секунди. Количеството трева в резервоара ще бъде 2. След това, той ще избере да изпразни резервоара за 2 секунди. На първата лента той прекарва общо 4 секунди.

След това, той ще мине през втората лента за 10 секунди и ще окоси 4 единици трева. Влад решава да не изпразни резервоара след тази лента. На втората лента той прекарва общо 10 секунди.

На третата лента Влад започва да коси. След 1 единица трева, резервоарът се напълва и той трябва да добути косачката до края, да изпразни резервоара, и отново да мине през третата лента, за да окоси останалите 5 единици трева. След окосяване на цялата ливада той отново трябва да изпразни резервоара. Така на третата лента той прекарва общо  $3 + 2 + 3 + 2 = 10$  секунди.

Общо окосяването на цялата ливада отнема  $4 + 10 + 10 = 24$  секунди. Може да бъде доказано, че това е оптимална стратегия за този тест.

## Пример 2

Да разгледаме следното извикване:

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

В този пример имаме 4 ленти, резервоарът е с капацитет 10, и отнема 4 секунди да бъде изпразнен.

Една оптимална стратегия е Влад да мине през първите 3 ленти, напълвайки резервоара по време на третата, и оставайки количеството неокосена трева в лентите  $[0, 0, 1, 7]$ . След изпразване на резервоара в края на лента 3, оставащите 2 ленти могат да бъдат окосени и резервоарът да бъде изпразнен още веднъж финално.

Общото нужно време е  $a[0] + a[1] + a[2] + b + a[2] + a[3] + b = 17$

## Локален грейдър

Локалният грейдър чете входа от стандартния вход в следния формат:

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

и извежда на стандартния изход резултата от извикването на `mow` със съответните параметри.