

## ბალახმკრეჭი

პოენარის ციხესიმაგრის თავგადასავლების შემდეგ ვლადი სახლში დაბრუნდა და როგორც ნამდვილ რუმინელს შეეფერება, პირველად მისი ცხენის გამოკვება გაახსენდა. ცხენი არც ისე პრეტენზიული მჭამელია, ამიტომ ვლადი მინდვრის ბალახს იყენებს მის გამოსაკვებად.

ვლადს აქვს ბალახის საკრეჭი მანქანა, რომლის ტევადობა არის  $c$ . მან გადაწყვიტა დაეყოს მისი მინდორი  $n$  ცალ ზოლად, ზოლები გადანომრილია 0-დან  $n - 1$ -ის ჩათვლით, რომლებიც მან ამავე თანმიმდევრობით უნდა გაკრიჭოს. ზოლი ნომრით  $i$  მოიცავს გაუკრეჭავ ბალახს  $v_i$  რაოდენობით და გაურკვეველი მიზეზების გამო ვლადს  $a_i$  წამი სჭირდება იმისათვის, რომ ამ ზოლზე ბალახის საკრეჭი მანქანა გადაატაროს.

რამდენიმე ზოლის გაკრეჭის შემდეგ მანქანის ტევადობა შესაძლოა შეივსოს, რა დროსაც ის ბალახის კრეჭას შეწყვეტს და ზოლზე გაუკრეჭავი ბალახი დარჩება. ყოველთვის როცა ეს ხდება, მანქანა უნდა დაიცალოს, რასაც  $b$  წამი სჭირდება და აუცილებლად ზოლის ბოლოს ხდება. თუ მანქანა გაივსო მაშინ, როცა ვლადი კრეჭდა ზოლს ნომრით  $i$ , მან უნდა მიაგოროს მანქანა ზოლის ბოლომდე, დაცალოს ბალახისგან და გააგრძელოს კიდევ ერთხელ (ან რამდენჯერაც დასჭირდება) ზოლის გაკრეჭა დარჩენილი ბალახისგან. მაგალითად თუ ზოლზე ნომრით  $i$  გვინევს მანქანის სამჯერ გადატარება, მაშინ დაგვებარდება  $a_i + b + a_i + b + a_i$  წამი. **მინდვრის მთლიანად გაკრეჭის შემდეგ აუცილებელია მანქანის დაცლა.**

ბევრი ფიქრის შემდეგ ვლადი მიხვდა, რომ ხანდახან დროის დაზოგვის მიზნით უკეთესია მანქანა დავცალოთ იქამდე, სანამ ის მთლიანად გაივსება. თუმცა მან ზუსტად არ იცის რა სტრატეგია გამოიყენოს. ამიტომაც ის თქვენ გთხოვთ დახმარებას.

მოცემულია ბალახის რაოდენობა თითოეულ ზოლზე, წამების რაოდენობა, რომელიც საჭიროა თითოეულ ზოლზე მანქანის გადასატარებლად, მანქანის ტევადობა და მისი დაცლის დრო. იპოვეთ მინდვრის გასათიბად საჭირო მინიმალური დრო.

## იმპლემენტაციის დეტალები

თქვენ უნდა დააიმპლემენტიროთ შემდეგი ფუნქცია.

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

- $n$ : ზოლების რაოდენობა მინდორში
- $c$ : მანქანის ტევადობა
- $b$ : მანქანის დაცლის დრო
- $a$ :  $n$  სიგრძის ვექტორი რომელიც აღნიშნავს თითოეულ ზოლზე მანქანის გადატარების დროს

- $v$ :  $n$  სიგრძის ვექტორი რომელიც აღნიშნავს თითოეულ ბოლზე ბალახის რაოდენობას
- ამ ფუნქციამ უნდა დააბრუნოს ერთადერთი რიცხვი, მინდვრის გასათიბად საჭირო მინიმალური დრო
- ეს პროცედურა ყოველი ტესტისთვის ერთხელ უნდა იყოს გამოძახებული

## შეზღუდვები

- $1 \leq n \leq 200\,000$
- $1 \leq a_i \leq 10^9$  (ყოველი  $i$ -სთვის სადაც  $0 \leq i < n$ )
- $1 \leq v_i \leq 10^9$  (ყოველი  $i$ -სთვის სადაც  $0 \leq i < n$ )
- $1 \leq b \leq 10^9$
- $1 \leq c \leq 10^9$
- გარანტირებულია რომ სწორი პასუხი იქნება მაქსიმუმ  $10^{18}$

## ქვეამოცანები

1. (9 ქულა) ყველა მოცემული რიცხვი არის მაქსიმუმ 200
2. (16 ქულა)  $n, c \leq 5000$  და  $v_i \leq 5000$  სადაც  $0 \leq i < n$
3. (36 ქულა)  $c \leq 200\,000$
4. (17 ქულა)  $a[0] = a[1] = \dots = a[n-1]$
5. (22 ქულა) არანაირი დამატებითი შეზღუდვა

## მაგალითები

### მაგალითი 1

განვიხილოთ შემდეგი გამოძახება:

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

ამ მაგალითში გვაქვს 3 ბოლი, მანქანის ტევადობაა 5 და დასაცლელად სჭირდება 2 წამი.

ვლადი გაკრეჭს პირველ ბოლს 2 წამში. გაკრეჭილი ბალახის რაოდენობა მანქანაში იქნება 2. შემდეგ ის დაცლის მანქანას 2 წამში. პირველი ბოლისთვის ის დახარჯავს 4 წამს.

შემდეგ ვლადი გადავა მეორე ბოლზე და გაკრეჭს 4 რაოდენობის ბალახს. ის გადაწყვეტს რომ არ დაცალოს მანქანა. ამ ბოლზე დახარჯული წამების რაოდენობა გამოვა 10.

შემდეგ ის იწყებს მესამე ბოლის კრეჭას. ერთი ბალახის გაკრეჭის შემდეგ მანქანა ივსება, ვლადს უნევს გავიდეს ბოლის ბოლოში, მანქანა დაცალოს და ბოლის გაკრეჭა გააგრძელოს. გაკრეჭვის შემდეგ მანქანა კიდევ ერთხელ უნდა დაცალოს. მესამე ბოლზე დახარჯული დრო გამოვა  $3 + 2 + 3 + 2 = 10$  წამი.

ჯამურად, ის დახარჯავს  $4 + 10 + 10 = 24$  წამს. მტკიცდება რომ ეს ოპტიმალური სტრატეგიაა.

### მაგალითი 2

განვიხილოთ შემდეგი გამოძახება:

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

ამ მაგალითში გვაქვს 4 ზოლი, მანქანის ტევადობაა 10 და დაცლისთვის სჭირდება 4 წამი.

ოპტიმალური სტრატეგიაა გადაატაროს მანქანა პირველ 3 ზოლს, რის შედეგადაც მანქანა გაივსება და ბალახის რაოდენობის ვექტორი იქნება შემდეგნაირი:  $[0, 0, 1, 7]$ . ამის შემდეგ მანქანა უნდა დაიცალოს, ბოლო ორი ზოლი გაიკრიჭოს და კიდევ ერთხელ დაიცალოს.

ჯამურად დახარჯული წამების რაოდენობა გამოვა  $a[0] + a[1] + a[2] + b + a[2] + a[3] + b = 17$ .

## Sample grader

Sample grader კითხულობს შემდეგი ფორმატით:

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

და გამოაქვს `mow` ფუნქციის გამოძახების შედეგი.