Rasenmäher

Nach seinen Abenteuern in der Festung Poenari kehrt Vlad nach Hause zurück, und wie es sich für einen echten Rumänen gehört, ist sein erster Gedanke, dass er sein Pferd füttern sollte. Das Pferd ist nicht sehr wählerisch, wenn es um Futter geht, also benutzt Vlad seinen Rasen als Hauptnahrungsquelle für das Pferd.

Für diese Aufgabe hat Vlad einen Rasenmäher, der einen Behälter mit Fassungsvermögen c hat. Er beschliesst, seinen Rasen in n Bahnen zu unterteilen, die von 0 bis $n-1$ nummeriert sind und die er in genau dieser Reihenfolge mähen muss. Jede Bahn i enthält $v[i]$ Einheiten an ungeschnittenem Gras, und Vlad braucht genau $a[i]$ Sekunden, um den Mäher über diese Bahn zu schieben.

Nach einigen Bahnen kann es sein, dass der Mäher sein volles Fassungsvermögen erreicht und aufhört Gras zu mähen, so dass etwas auf der Bahn liegen bleibt. Jedes Mal, wenn das passiert, muss der Behälter mit dem Gras geleert werden, was b Sekunden dauert und nur am Ende einer Bahn möglich ist. D.h., wenn der Behälter voll wird, während Vlad über Bahn i fährt, muss er den Rasenmäher bis zum Ende der Bahn schieben, den Behälter leeren und dann noch einmal (bzw. so oft wie notwendig) über die Bahn fahren, um das übrig gebliebene Gras zu mähen. Wenn z. B. eine Bahn 3 Mal durchfahren werden muss, um das gesamte Gras zu mähen, dauert das $a[i] + b + a[i] + b + a[i]$ Sekunden. **Nach dem Mähen des gesamten Rasens muss der Behälter geleert werden.**

Nach langem Nachdenken und Klagen darüber, dass er viel zu lange braucht, um mit dem Mähen fertig zu werden, ist Vlad zu dem Schluss gekommen, dass es manchmal zeitsparender wäre, den Behälter zu leeren, bevor er voll ist, aber er ist sich nicht sicher, welche Strategie er am besten anwenden sollte. Deshalb bittet er dich um deine Hilfe.

Finde den besten Weg für Vlad um den gesamten Rasen in minimaler Zeit zu mähen.

Implementierungsdetails

Implementiere die folgende Funktion

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

- n: Anzahl an Bahnen, die der Rasen hat
- c: Fassungsvermögen des Behälters vom Rasenmäher
- b: Zeit in Sekunden, die es braucht um den Behälter zu leeren
- a: ein Vektor der Länge n, welcher die Zeit angibt, die man braucht um einmal über jede Bahn zu gehen
- v: ein Vektor der Länge n, welcher die Menge an Gras auf jeder Bahn angibt
- Die Funktion sollte eine Zahl zurückgeben, die minimale Zeit um den gesamten Rasen zu mähen
- Die Funktion wird genau einmal für jeden Testfall aufgerufen

Beschränkungen

- $1 \leq n \leq 200\,000$
- $1 \leq a[i] \leq 10^9$ (für jedes i sodass $0 \leq i < n$)
- $1 \leq v[i] \leq 10^9$ (für jedes i sodass $0 \leq i < n$)
- $1 \leq b \leq 10^9$
- $1 \leq c \leq 10^9$
- Es ist garantiert, dass das korrekte Ergebnis maximal 10^{18} ist.

1. (9 Münzen) Alle gegebenen Werte ($n, b, c, a[i]$ and $v[i]$) sind maximal 200
2. (16 Münzen) $n, c \leq 5000$ und $v[i] \leq 5000$ für alle $0 \leq i < n$
3. (36 Münzen) $c \leq 200\,000$
4. (17 Münzen) $a[0] = a[1] = \ldots = a[n-1]$
5. (22 Münzen) Keine weiteren Beschränkungen

Beispiele

Beispiel 1

Betrachte den folgenden Funktionsaufruf

```
mow(3, 5, 2, [2, 10, 3], [2, 4, 6])
```

In diesem Beispiel gibt es 3 Bahnen, der Behälter hat ein Fassungsvermögen von 5 und es dauert 2 Sekunden, ihn zu leeren.

Für dieses Beispiel wird Vlad die erste Bahn in 2 Sekunden mähen. Die Menge des Grases im Behälter beträgt 2. Dann leert er den Behälter in 2 Sekunden. Für die erste Bahn benötigt er 4 Sekunden.

Dann mäht er die zweite Bahn. Er mäht 4 Einheiten Gras. Er beschliesst, den Behälter nach Beendigung der zweiten Bahn nicht zu leeren. Der Zeitaufwand für die zweiten Bahn beträgt also 10 Sekunden.

Anschliessend beginnt er die dritte Bahn zu mähen. Nach einer Einheit Gras ist sein Behälter voll, also muss er bis zum Ende der Bahn fahren, den Behälter leeren und dann erneut mit dem Mähen der dritten Bahn beginnen. Denk daran, dass der Behälter geleert werden muss, nachdem der gesamte Rasen gemäht wurde. Der Zeitaufwand für die dritte Bahn beträgt $3 + 2 + 3 + 2 = 10$ Sekunden.

Insgesamt braucht er also $4 + 10 + 10 = 24$ Sekunden. Es kann bewiesen werden, dass dies die optimale Strategie ist, die Vlad zum Mähen des Rasens verwendet.

Beispiel 2

Betrachte den folgenden Funktionsaufruf

```
mow(4, 10, 4, [1, 2, 1, 4], [3, 2, 6, 7])
```

In diesem Beispiel gibt es 4 Bahnen, der Behälter hat ein Fassungsvermögen von 10 und es dauert 4 Sekunden, ihn zu leeren.

Die optimale Strategie besteht darin, nur die ersten 3 Bahnen zu mähen. Im Anschluss ist der Behälter voll und der Vektor der Grasmengen ist `[0, 0, 1, 7]`. Danach wird der Behälter geleert, die letzten 2 Bahnen werden gemäht und am Ende wird der Behälter wieder geleert.

Die Gesamtzahl der Sekunden beträgt $a[0] + a[1] + a[2] + b + a[2] + a[3] + b = 17$.

Beispiel-Grader

Der Beispiel-Grader liest die Eingabe in folgendem Format:

- Zeile 1: $n\ c\ b$
- Zeile 2: $a[0]\ a[1] \ldots a[n-1]$
- Zeile 3: $v[0]\ v[1] \ldots v[n-1]$

und gibt das Ergebnis des Aufrufs von `mow` mit den entsprechenden Parametern aus.