Mario - Der Sammler

Die Figur Mario (eine fiktive, selbst ausgedachte Figur) hat in ihrem neuesten Spiel die Aufgabe alle Münzen in möglichst kurzer Zeit zu sammeln.

Das Spiel besteht aus n Leveln, nummeriert von 0 bis $n-1$, welche in dieser Reihenfolge gespielt werden müssen. Mario verfügt zum Sammeln über eine Geldtasche, die c Münzen fassen kann. Level i enthält genau $v[i]$ Münzen und Mario braucht $a[i]$ Sekunden, um es zu absolvieren.

Mario will alle Münzen in seinem geheimen Safe zusammentragen. Dafür muss er sie immer wieder von seiner Geldtasche in den Safe transferieren, was ihn jeweils b Sekunden kostet. Dies kann er entweder zu einem selbstgewählten Zeitpunkt (bspw. zwischen zwei Leveln) machen oder er muss es gezwungenermaßen tun, nämlich wenn seine Geldtasche voll ist. Passiert dies innerhalb eines Levels, muss er zuerst das Level fertig spielen (ohne weitere Münzen sammeln zu können), die gesammelten Münzen in den Safe transferieren und anschließend das Level wieder von vorn beginnen. Bei erneutem Beginnen eines Levels sind nur mehr die nicht-eingesammelten Münzen in selbigem vorhanden. Sollte beispielsweise Mario Level i 3-mal durchlaufen müssen, um alle Münzen einzusammeln, braucht er dafür $a[i]+b+a[i]+b+a[i]$ Sekunden. **Nicht vergessen: Nach Ende des Spiels muss Mario noch ein letztes Mal seine Münzen in den Safe legen.**

Um den Speedrun richtig zu machen, braucht er deine Hilfe herauszufinden, wann er am Besten seine Geldtasche leert. Hierfür gegeben sind die Anzahl an Münzen pro Level, die Zeit die Mario für dieses Level braucht, die Größe der Geldtasche und die Zeit die er für das Leeren selbiger benötigt. Finde die minimale Zeit, die Mario braucht, um alle Münzen einzusammeln.

Implementationsdetails

Du sollst folgende Funktion implementieren:

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

- n: Anzahl der Level
- c: Größe der Geldtasche
- b: Zeit, die Mario zum Leeren der Geldtasche braucht
- a: ein `vector` der Länge n, mit der Zeit die Mario pro Level braucht
- v: ein `vector` der Länge n, mit der Anzahl der Münzen pro Level
- Die Methode soll einen Integer zurückgeben, welcher die minimale Zeit angibt, um alle Münzen zu sammeln.
- Die Methode wird pro Testfall genau einmal aufgerufen.

('mow' ist in diesem Fall rumänisch für 'sammeln' - gleich wie 'Mario - der Sammler' die deutsche Version von 'Landmower' ist.)

Einschränkungen

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

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

Beispiele

Beispiel 1

Gegeben folgender Funktionsaufruf:

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

Es gibt 3 Level, die Geldtasche kann bis zu 5 Münzen fassen und es braucht 2 Sekunden um sie zu leeren.

In diesem Testfall wird Mario das erste Level in 2 Sekunden spielen und dessen 2 Münzen eingesammeln. Anschließend leert er die Geldtasche. So braucht er für das erste Level insgesamt 4 Sekunden.

Danach spielt er das zweite Level. Er sammelt 4 Münzen und entscheidet sich gegen ein erneutes Leeren $\rightarrow$ er braucht 10 Sekunden insgesamt.

Im dritten Level muss er die Geldtasche leeren nachdem er die erste Münze aufgesammelt hat. Er spielt bis zum Ende weiter, leet sie und spielt das Level erneut. Hier ist zu beachten, dass nach Ende des Spiels die Geldtasche erneut geleert werden muss. So braucht er für Level 3 insgesammt $3 + 2 + 3 + 2 = 10$ Sekunden.

Alles zusammen braucht er $4 + 10 + 10 = 24$ Sekunden. Es kann bewiesen werden, dass dies die optimale Strategie ist.

Beispiel 2

Gegeben folgender Funktionsaufruf:

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

In diesem Beispiel gibt es 4 Level, die Geldtasche ist 10 groß und es braucht 4 Sekunden sie zu leeren.

Die optimale Strategie besteht darauf die ersten 3 Level durchzumachen, was zu einer vollen Geldtasche führt. Der `vector` mit den Münz-Anzahlen schaut nun so aus: `[0, 0, 1, 7]`. Danach leert Mario seine Geldtasche, spielt Level 3 erneut und anschließend Level 4. Zum Schluss muss er nochmals die Geldtasche leeren.

Die Gesamt-Anzahl an Sekunden ist so $a[0] + a[1] + a[2] + b + a[2] + a[3] + b = 17$.

Beispiel Grader

Der Beispiel Grader liest den Input im folgeden 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 nach dem Aufruf von `mow` im entsprechenden Format aus.