PROBLEM 4

Backpacking

Input file: `backin.txt`
Output file: `backout.txt`

Time and memory limits: 1 second, 1 GB

Norman is planning a backtracking trip through N towns, numbered 1 to N in the order that he will visit them. After careful planning, he knows that it will take D_i days to travel from town i to town $i+1$. Norman will start in town 1 and backpack all the way to town N, travelling for a total of $D_1 + D_2 + \ldots + D_{N-1}$ days.

Norman's backpack can fit up to K cans of food, and each day he will eat exactly one can. The price of food varies in each town; in town i, food costs C_i dollars per can. He can buy as many cans as he wants from each town as long as the total number of uneaten cans doesn't exceed K.

Norman will start in town 1 with an empty backpack, and wants to spend as little as possible on food by strategically choosing how much he buys in each town. What is the minimum total amount that he must spend to reach town N?

Subtasks and constraints

Your program will be graded using many secret tests. Every test follows some rules:

- $2 \leq N \leq 200\,000$.
- $1 \leq K \leq 1\,000\,000$.
- $1 \leq D_i \leq K$ for all i. That is, Norman's backpack can store enough food for him to travel between consecutive towns on his trip.
- $D_1 + D_2 + \ldots + D_{N-1} \leq 1\,000\,000$. That is, Norman's trip is at most $1\,000\,000$ days in total.
- $1 \leq C_i \leq 20$ for all i.

The secret tests are divided into subtasks. Your program must correctly solve **every test** within a subtask to earn the marks for that subtask:

- For Subtask 1 (20 marks), $C_i \leq C_{i+1}$ for all i. That is, the towns are ordered from cheapest to most expensive.
- For Subtask 2 (30 marks), $K = 1\,000\,000$.
- For Subtask 3 (30 marks), $N \leq 1000$.
- For Subtask 4 (20 marks), no special rules apply.

Input

Your program must read input from the file `backin.txt`. When testing on your own computer, this file must be placed in the same folder as your program. We strongly recommend using the solution templates (which you can find on the *Templates & Downloads* page of the competition website) to help you with input and output.

The file `backin.txt` follows a specific format:

- The 1st line of input contains the integers N and K.
- The 2nd line of input contains $N-1$ integers describing the distances between towns. The ith of these is D_i.
- The 3rd line of input contains N integers describing the cost of food in each town. The ith of these is C_i.

Output

Your program must write a single integer to the file `backout.txt`: the minimum total amount (in dollars) that Norman must spend to reach town N.

Sample input 1

```
3 5
4 3
2 3 4
```

Sample input 2

```
5 1000000
2 2 2 2
5 3 4 1 2
```

Sample input 3

```
5 3
2 2 2 2
5 3 4 1 2
```

Sample output 1

```
16
```

Sample output 2

```
24
```

Sample output 3

```
25
```

Explanation

- In the 1st sample case, Norman's best strategy is:
  - fill his backpack by buying 5 cans at town 1 for $5 \times 2 = 10$ dollars
  - backpack from town 1 to town 2, which takes 4 days; when he arrives at town 2 he has 1 can left in his backpack
  - buy 2 cans at town 2 for $2 \times 3 = 6$ dollars; he now has 3 cans
  - backpack from town 2 to town 3, which takes 3 days; he arrives at town 3 with no cans left.

  This costs a total of $10 + 6 = 16$ dollars.
- In the 2nd sample case, Norman's best strategy is:
  - buy 2 cans at town 1 for $2 \times 5 = 10$ dollars
  - backpack from town 1 to town 2, arriving with no cans left in his backpack
  - buy 4 cans at town 2 for $4 \times 3 = 12$ dollars
  - backpack from town 2 to town 3, arriving with 2 cans left in his backpack
  - don't buy any cans at town 3
  - backpack from town 3 to town 4, arriving with no cans left in his backpack
  - buy 2 cans at town 4 for $2 \times 1 = 2$ dollars
  - backpack from town 4 to town 5, arriving with no cans left in his backpack.

  This costs a total of $10 + 12 + 2 = 24$ dollars.
- In the 3rd sample case, Norman's best strategy is:
  - buy 2 cans at town 1 for $2 \times 5 = 10$ dollars
  - backpack from town 1 to town 2, arriving with no cans left in his backpack
  - buy 3 cans at town 2 for $3 \times 3 = 9$ dollars; his backpack is now full
  - backpack from town 2 to town 3, arriving with 1 can left in his backpack
  - buy 1 can at town 3 for $1 \times 4 = 4$ dollars; he now has 2 cans
  - backpack from town 3 to town 4, arriving with no cans left in his backpack
  - buy 2 cans at town 4 for $2 \times 1 = 2$ dollars
  - backpack from town 4 to town 5, arriving with no cans left in his backpack.

  This costs a total of $10 + 9 + 4 + 2 = 25$ dollars.