

Aggiornamento della macchina virtuale (vm)

Georgio Audritto is updating the OII virtual machine. It contains N programs, numbered from 0 to $N - 1$: the version of the i -th one is currently A_i . The sequence A_i is nondecreasing (i.e. $A_i \leq A_{i+1}$ for each $0 \leq i \leq N - 2$).

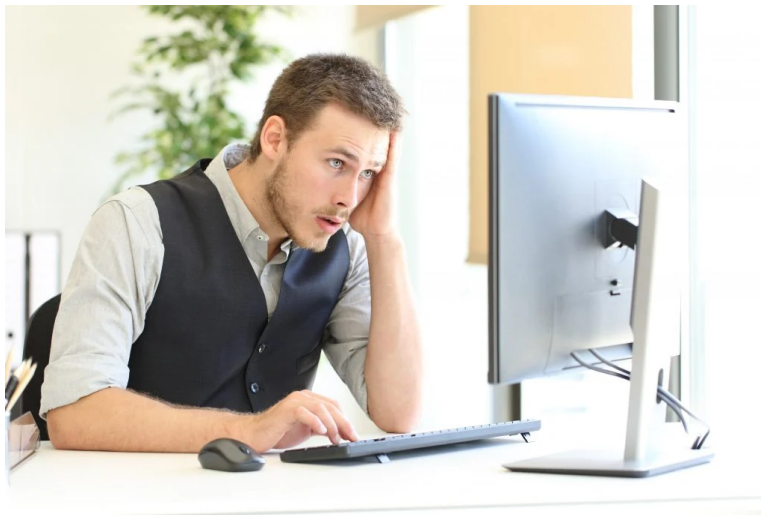


Figure 1: Georgio while he is updating the virtual machine


For each update, Georgio chooses a program i and updates it by B_i versions (so A_i increases by B_i). The same program can be updated multiple times. Moreover:

- Georgio can't update the program number $N - 1$.
- The i -th program depends on all the previous ones, so it can't have a lower version than the programs with lower index (for backward compatibility). In other words, after each update, the sequence A_i must be nondecreasing ($A_i \leq A_{i+1}$ must hold for each $0 \leq i \leq N - 2$). Note that the order of the programs doesn't change after an update.

What's the maximum number of updates that Georgio can make?

Implementation

You should submit a single file, with a `.cpp` extension.

 Among the attachments in this task you will find a template `vm.cpp` with a sample implementation.

You will have to implement the following function:

```
C++ | long long aggiorna(int N, vector<int> A, vector<int> B);
```

- The integer N represents the number of programs in the virtual machine.
- The vector A contains the initial version of each program. In particular, for each $0 \leq i \leq N - 1$, A_i is the version of the i -th program.

- The vector B specifies how the programs are updated. In particular, for each $0 \leq i \leq N - 2$, A_i increases by B_i after an update of the i -th program.

The function `aggiorna` must return the maximum number of updates that Georgio can make.

Grader di prova

Among this task's attachments you will find a simplified version of the grader used during the evaluation, which you can use to test your solutions locally. The sample grader reads data from `stdin`, calls the functions that you should implement and writes back on `stdout` using the following format.

The input file consists of 3 lines, containing:

- Line 1: the integer N .
- Line 2: the N integers A_0, \dots, A_{N-1} .
- Line 3: the $N - 1$ integers B_0, \dots, B_{N-2} .

The output file consists of a single line, containing the value returned by `aggiorna`.

Constraints

- $2 \leq N \leq 2 \cdot 10^5$.
- $1 \leq A_i, B_i \leq 10^9$.
- $A_i \leq A_{i+1}$ for each $0 \leq i \leq N - 2$.

Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.

- **Subtask 1 [0 points]:** Examples.
- **Subtask 2 [12 points]:** $B_i = 1$
- **Subtask 3 [20 points]:** $N \leq 500, A_i \leq 200$
- **Subtask 4 [22 points]:** $A_i \leq 200$
- **Subtask 5 [19 points]:** $N = 2 \cdot 10^5$, the input is chosen uniformly randomly among all the valid inputs.
- **Subtask 6 [27 points]:** No additional limitations.

Examples

stdin	stdout
6 2 7 11 14 18 25 8 4 6 3 4	7
8 3 27 86 95 419 852 1473 6461 100 100 100 1000 1000 1000 10000	0

Explanations

In the **first example**, Georgio can make the following updates:

- Update of the program 3: the versions become [2, 7, 11, 17, 18, 25];
- Update of the program 2: the versions become [2, 7, 17, 17, 18, 25];
- Update of the program 1: the versions become [2, 11, 17, 17, 18, 25];
- Update of the program 0: the versions become [10, 11, 17, 17, 18, 25];
- Update of the program 1: the versions become [10, 15, 17, 17, 18, 25];
- Update of the program 4: the versions become [10, 15, 17, 17, 22, 25];
- Update of the program 3: the versions become [10, 15, 17, 20, 22, 25].

7 updates have been made: it can be shown that it's the maximum possible for this input.

In the **second example** Georgio can't update any program.