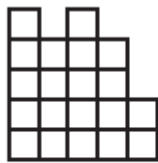


Escada perfeita

Uma construtora, durante a criação de um parque temático, encontrou no terreno um conjunto de várias pilhas de cubos de pedra. Em vez de pagar pela remoção dos cubos de pedras, um dos arquitetos da empresa achou interessante utilizar as pedras para decoração do parque, determinando que as pedras fossem rearranjadas no formato de “escada”. Para isso, os funcionários deveriam mover alguns cubos para formar os degraus das escadas. Só que o arquiteto decidiu que, entre uma pilha e outra de pedras deveria haver exatamente uma pedra e diferença, formando o que ele chamou de escada perfeita. O exemplo abaixo mostra um conjunto de cinco pilhas de pedras encontradas e as cinco pilhas como ficaram após a arrumação em escada perfeita.



Arranjo de pilhas de
cubos de pedras encontrado



Escada perfeita construída
após o movimento de alguns cubos

Tarefa

Dada uma sequência de pilhas de cubos de pedras com suas respectivas alturas, você deve determinar o número mínimo de pedras que precisam ser movidas para formar uma escada perfeita com exatamente o mesmo número de pilhas de pedras encontrado inicialmente (ou seja, não devem ser criadas ou eliminadas pilhas de pedras). O degrau mais baixo da escada deve sempre estar do lado esquerdo. Dica: a somatório $1 + 2 + 3 + \dots + n = n * (n + 1) / 2$

Entrada

A entrada contém um único conjunto de testes, que deve ser lido do dispositivo de entrada padrão. A primeira linha contém um inteiro N que indica o número de pilhas de pedras. A segunda linha contém N números inteiros positivos que indicam a quantidade de cubos de pedras em cada uma das pilhas, da esquerda para a direita.

Saída

Seu programa deve imprimir, na saída padrão, uma única linha, contendo um inteiro: o número mínimo de cubos de pedras que devem ser movidos para transformar o conjunto de pilhas em uma escada perfeita, conforme calculado pelo seu programa. Caso não seja possível efetuar a transformação em escada perfeita, imprima como resultado o valor -1.

Exemplos

Entrada 5 5 4 5 4 2 Saída 5	Entrada 6 9 8 7 6 5 4 Saída 9	Entrada 2 1 5 Saída -1
---	---	--