

Șir frumos

Ioana este pasionată de tot felul de șiruri numerice. Analizând mai multe șiruri, ea a identificat un șir care i-a plăcut în special, și l-a numi „Șir frumos”. Un șir cu n elemente – a_1, a_2, \dots, a_n se numește frumos, dacă există două numere întregi a și b , astfel încât $a_i = a + b \cdot i$ pentru orice $i, 1 \leq i \leq n$. Astfel, pentru orice șir b_1, b_2, \dots, b_n Ioana aplică următoarele două operații pentru a transforma-o într-un Șir frumos:

- (1) Se alege un indice p și se incrementează cu 1 primele p elemente. Urmare a aplicării acestei operații se obține șir nou c_1, c_2, \dots, c_n , unde $c_i = b_i + 1$ pentru $i \leq p$ și $c_i = b_i$ pentru $i > p$.
- (2) Se alege un indice p și se decrementează cu 1 primele p elemente. Urmare a aplicării acestei operații se obține șir nou c_1, c_2, \dots, c_n , unde $c_i = b_i - 1$ pentru $i \leq p$ și $c_i = b_i$ pentru $i > p$.

De exemplu, fie șirul $[1, 4, 2, 3]$. Dacă aplicăm operația (1) pentru $p = 2$, vom obține șirul $[2, 5, 2, 3]$. Alternativ, dacă aplicăm operația (2) pentru $p = 4$, vom obține șirul $[0, 3, 1, 2]$.

Sarcină: Elaborați un program, care având un șir determină numărul minim de operații necesare pentru a-l transforma într-un șir frumos.

Date de intrare: Prima linie a intrării standard conține un număr întreg n – numărul de elemente din șir. A doua linie a intrării standard conține n numere întregi b_1, b_2, \dots, b_n elementele șirului inițial.

Date de ieșire: Ieșirea standard va conține un singur număr întreg – numărul minim de operații necesare pentru a transforma șirul într-un șir frumos.

Restricții: $2 \leq n \leq 10^5$; $-10^9 \leq a_i \leq 10^9, i = 1, 2, \dots, n$. Restricțiile referitoare la timpul de execuție și volumul utilizat de memorie sunt date în descrierea generală a problemelor propuse pentru rezolvare. Fișierul sursă va avea denumirea `sir.pas`, `sir.c` sau `sir.cpp`.

Punctare: Testele vor fi împărțite în 5 grupuri, după cum urmează:

Grup	Procentaj teste	Restricții
1 (Subtask 1)	10%	$n \leq 3$
2 (Subtask 2)	10%	$n \leq 1000, a_i \leq 100$
3 (Subtask 3)	10%	$ a_i \leq 100$
4 (Subtask 4)	20%	$n \leq 1000$
5 (Subtask 5)	50%	Fără restricții adiționale

Exemplu 1:

Intrare

```
3
2 1 3
```

Ieșire

```
3
```

Exemplu 2:

Intrare

```
4
1 1 -1 -2
```

Ieșire

```
2
```

Explicații:

În primul exemplu, unul din modurile de a obține un șir frumos folosind 3 operații este:

1) Operația (2) cu $p = 1$. Obținem șirul $[1,1,3]$.

2) Operația (1) cu $p = 2$. Obținem șirul $[2,2,3]$.

3) Operația (2) cu $p = 1$. Obținem șirul $[1,2,3]$.

Șirul obținut este frumos pentru că există a, b astfel încât $a = 0, b = 1$.

În al doilea exemplu, unul din modurile de a obține un șir frumos folosind 2 operații este:

1) Operația (2) cu $p = 2$. Obținem șirul $[0,0,-1,-2]$.

2) Operația (1) cu $p = 1$. Obținem șirul $[1,0,-1,-2]$.

Șirul obținut este frumos pentru că există a, b astfel încât $a = 2, b = -1$.