



Mengatur Sepatu

Adnan memiliki toko sepatu terbesar di Baku. Sebuah kotak berisi n pasang sepatu baru saja tiba di toko. Setiap pasang berisi dua buah sepatu dengan ukuran yang sama: sepatu kiri dan sepatu kanan. Adnan telah meletakkan semua $2n$ buah sepatu dalam satu baris yang terdiri dari $2n$ **posisi** yang dinomori 0 sampai dengan $2n - 1$ dari kiri ke kanan.

Adnan ingin mengatur ulang sepatu-sepatu tersebut ke dalam suatu **pengaturan yang benar**. Suatu pengaturan disebut benar jika dan hanya jika untuk setiap i ($0 \leq i \leq n - 1$), kondisi-kondisi berikut terpenuhi:

- Sepatu-sepatu pada posisi $2i$ dan $2i + 1$ berukuran sama.
- Sepatu pada posisi $2i$ adalah sepatu kiri.
- Sepatu pada posisi $2i + 1$ adalah sepatu kanan.

Untuk tujuan ini, Adnan dapat melakukan serangkaian penukaran. Dalam setiap penukaran, ia memilih dua buah sepatu yang **bersebelahan** pada saat itu dan menukarnya (maksudnya, mengambil kedua sepatu tersebut dan menempatkan masing-masing pada posisi sepatu yang lainnya). Dua sepatu disebut bersebelahan jika posisinya berbeda satu.

Tentukan banyak minimal penukaran yang perlu dilakukan Adnan untuk mencapai pengaturan sepatu-sepatu yang benar.

Rincian implementasi

Anda harus melakukan implementasi dari prosedur berikut:

```
int64 count_swaps(int[] S)
```

- S : sebuah array berisikan $2n$ buah bilangan bulat. Untuk setiap i ($0 \leq i \leq 2n - 1$), $|S[i]|$ adalah sebuah nilai bukan nol yang merupakan ukuran sepatu yang awalnya diletakkan pada posisi i . Di sini, $|x|$ menyatakan nilai absolut dari x , yang sama dengan x jika $x > 0$ dan sama dengan $-x$ jika $x < 0$. Jika $S[i] < 0$, sepatu pada posisi i adalah sepatu kiri; sebaliknya, adalah sepatu kanan.
- Prosedur ini harus mengembalikan banyaknya minimal penukaran (dari sepatu yang bersebelahan) yang perlu dilakukan untuk mencapai pengaturan yang benar.

Contoh

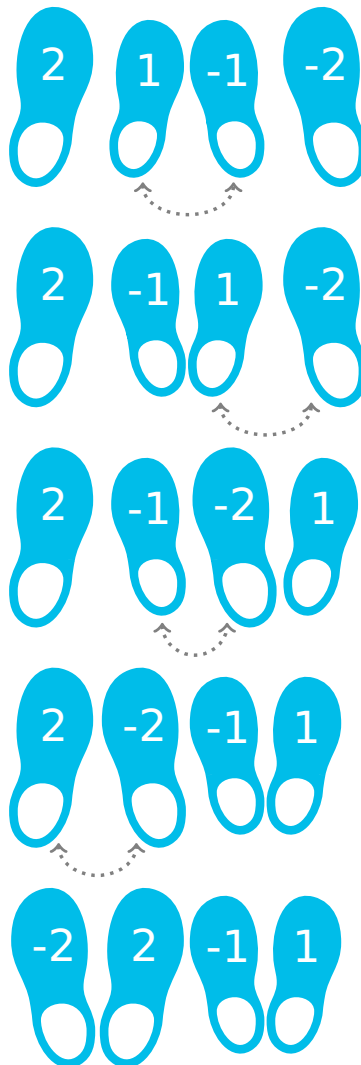
Contoh 1

Perhatikan panggilan berikut:

```
count_swaps([2, 1, -1, -2])
```

Adnan dapat mencapai pengaturan yang benar dengan 4 penukaran.

Misalnya, pertama ia dapat menukar sepatu 1 dan -1 , kemudian 1 dan -2 , kemudian -1 dan -2 , dan akhirnya 2 dan -2 . Ia kemudian akan mencapai pengaturan yang benar berikut: $[-2, 2, -1, 1]$. Tidak mungkin mencapai pengaturan yang benar dengan kurang dari 4 penukaran. Oleh karena itu, prosedur harus mengembalikan nilai 4.



Contoh 2

Pada contoh berikut, semua sepatu memiliki ukuran yang sama:

```
count_swaps([-2, 2, 2, -2, -2, 2])
```

Adnan dapat menukar sepatu-sepatu pada posisi 2 dan 3 untuk mencapai pengaturan yang benar $[-2, 2, -2, 2, -2, 2]$, maka prosedur harus mengembalikan nilai 1.

Batasan

- $1 \leq n \leq 100\,000$
- Untuk setiap i ($0 \leq i \leq 2n - 1$), $1 \leq |S[i]| \leq n$.
- Pengaturan sepatu yang benar dapat dicapai dengan melakukan serangkaian penukaran.

Subsoal

1. (10 poin) $n = 1$
2. (20 poin) $n \leq 8$
3. (20 poin) Semua sepatu berukuran sama.
4. (15 poin) Semua sepatu pada posisi $0, \dots, n - 1$ adalah sepatu kiri, dan semua sepatu pada posisi $n, \dots, 2n - 1$ adalah sepatu kanan. Juga, untuk setiap i ($0 \leq i \leq n - 1$), sepatu pada posisi i dan $i + n$ berukuran sama.
5. (20 poin) $n \leq 1000$
6. (15 poin) Tidak ada batasan tambahan.

Grader contoh

Grader contoh membaca masukan dengan format berikut:

- baris 1: n
- baris 2: $S[0] S[1] S[2] \dots S[2n - 1]$

Grader contoh mengeluarkan satu baris berisi nilai yang dikembalikan oleh `count_swaps`.