Menghitung Jamur (mushrooms)

Andrew, seorang pakar jamur, sedang menginvestigasi jamur asal Singapura.

Sebagai bagian dalam risetnya, Andrew mengumpulkan n jamur yang dinomori dari 0 sampai $n-1$. Tiap jamur merupakan salah satu dari dua spesies, yang disebut dengan A dan B.

Andrew tahu bahwa **jamur 0 digolongkan dalam spesies A**, akan tetapi karena kedua spesies terlihat sama, dia tidak mengetahui spesies dari jamur 1 sampai $n-1$.

Untungnya, Andrew memiliki sebuah mesin di lab nya yang dapat membantu dengan hal ini. Untuk menggunakan mesin ini, dua atau lebih jamur harus diletakkan dalam satu baris di dalam mesin (dalam urutan mana saja) dan mesin lalu dinyalakan. Kemudian, mesin tersebut menghitung banyaknya pasangan jamur **bersebelahan** dengan spesies berbeda. Sebagai contoh, apabila Anda meletakkan jamur dengan spesies $[A, B, B, A]$ (dengan urutan tersebut) ke dalam mesin, hasil penghitungan mesin adalah 2.

Akan tetapi, karena pengoperasian mesin sangatlah mahal, mesin hanya dapat digunakan beberapa kali. Selain itu, total banyaknya jamur yang diletakkan di dalam mesin dari semua penggunaan tidak dapat melebihi $100\ 000$. Gunakan mesin ini untuk membantu Andrew menghitung banyaknya jamur dengan spesies A.

Detail Implementasi

Anda harus mengimplementasikan fungsi berikut:

```
int count_mushrooms(int n)
```

- n: banyaknya jamur yang dikumpulkan Andrew.
- Funsi ini dipanggil tepat sekali, dan harus mengembalikan banyaknya jamur dengan spesies A.

Fungsi di atas dapat memanggil fungsi berikut:

```
int use_machine(int[] x)
```

- x: sebuah array dengan panjang antara 2 sampai n inklusif, yang mendeskripsikan label dari jamur yang diletakkan di mesin tersebut, secara berurutan.
- Elemen dari x harus merupakan bilangan bulat **berbeda** di antara 0 sampai $n-1$ inklusif.
- Didefinisikan d sebagai ukuran dari array x. Maka, fungsi ini harus mengembalikan banyaknya

index berbeda j, sehingga $0 \leq j \leq d-2$ dan jamur $x[j]$ dan $x[j+1]$ memiliki spesies yang berbeda.
- Fungsi ini dapat dipanggil paling banyak $20\ 000$ kali.
- Total banyaknya elemen dari x yang diberikan ke fungsi `use_machine` dari semua pemanggilan tidak boleh melebihi $100\ 000$.

Contoh

Contoh 1

Perhatikan skenario yang mana terdapat 3 jamur dengan spesies $[A, B, B]$, secara berurutan. Fungsi `count_mushrooms` akan dipanggil dengan cara berikut:

```
count_mushrooms(3)
```

Fungsi ini dapat memanggil `use_machine([0, 1, 2])`, yang (di skenario ini) mengembalikan 1. Kemudian fungsi ini dapat memanggil `use_machine([2, 1])`, yang mengembalikan 0.

Pada saat ini, terdapat cukup informasi untuk menyimpulkan bahwa hanya terdapat 1 jamur dengan spesies A. Jadi, fungsi `count_mushrooms` harus mengembalikan 1.

Contoh 2

Perhatikan skenario yang mana terdapat 4 jamur dengan spesies $[A, B, A, A]$, secara berurutan. Fungsi `count_mushrooms` akan dipanggil dengan cara berikut:

```
count_mushrooms(4)
```

Fungsi ini dapat memanggil `use_machine([0, 2, 1, 3])`, yang mengembalikan 2. Kemudian fungsi ini dapat memanggil `use_machine([1, 2])`, yang mengembalikan 1.

Pada saat ini, terdapat cukup informasi untuk menyimpulkan bahwa terdapat 3 jamur dengan spesies A. Jadi, fungsi `count_mushrooms` harus mengembalikan 3.

Batasan

- $2 \leq n \leq 20\ 000$

Penilaian

Apabila pada kasus uji manapun, pemanggilan ke fungsi `use_machine` tidak memenuhi aturan di atas, atau nilai pengembalian dari `count_mushrooms` salah, skor dari solusi Anda adalah 0. Selain itu, didefinisikan Q sebagai maksimum banyaknya pemanggilan ke fungsi `use_machine` dari

semua kasus uji. Maka, skor akan dihitung berdasarkan tabel berikut:

Kondisi	Skor
$20\,000 < Q$	0
$10\,010 < Q \leq 20\,000$	10
$904 < Q \leq 10\,010$	25
$226 < Q \leq 904$	$\frac{226}{Q} \cdot 100$
$Q \leq 226$	100

Pada beberapa kasus uji, perbuatan dari *grader* bersifat adaptif. Hal ini berarti bahwa di kasus uji tersebut, *grader* tidak memiliki barisan spesies jamur yang tetap. Sebagai gantinya, jawaban yang diberikan *grader* bergantung pada pemanggilan `use_machine` sebelumnya. Meskipun demikian, dijamin bahwa *grader* menjawab sedemikian sehingga setelah tiap interaksi terdapat paling tidak satu barisan spesies jamur yang konsisten dengan semua jawaban yang telah diberikan sebelumnya.

Contoh *grader*

Contoh *grader* membaca array s berisi bilangan bulat yang menyatakan spesies-spesies jamur. Untuk setiap $0 \leq i \leq n-1$, $s[i] = 0$ berarti spesies dari jamur i adalah A, sedangkan $s[i] = 1$ berarti spesies dari jamur i adalah B,
Contoh *grader* membaca masukan dengan format sebagai berikut:

- baris 1: n
- baris 2: $s[0]$ $s[1]$ $\ldots$ $s[n-1]$

Keluaran dari contoh *grader* akan dalam format sebagai berikut:

- baris 1: nilai yang dikembalikan `count_mushrooms`.
- baris 2: banyaknya pemanggilkan dari `use_machine`.

Perhatikan bahwa contoh *grader* tidak adaptif.