Funghi velenosi (mushrooms)

Andrew, massimo esperto di funghi a Singapore, ha raccolto n funghi numerati da 0 a $n-1$, ognuno dei quali può essere o della specie A (mortale) o della specie B (deliziosa). L'assistente di Andrew ha verificato che **il fungo 0 è di tipo A**, però le due specie sono molto simili, quindi non si conosce ancora la specie dei funghi da 1 a $n-1$. Ora Andrew deve cavarsela da solo!

Fortunatamente, nel suo laboratorio ha uno strumento che, una volta inseriti due o più funghi in fila al suo interno, è in grado di calcolare il numero delle coppie **adiacenti** che sono di specie diverse. Per esempio, inserendo funghi di tipo $[A, B, B, A]$ (in quest'ordine) il risultato sarà 2.

Utilizzare il macchinario è molto costoso, e può essere acceso solo un numero limitato di volte prima di rompersi. Inoltre, non è possibile superare i $100\,000$ funghi analizzati in totale tra tutte le operazioni. Aiuta Andrew a contare quanti sono i funghi di tipo A tramite questo macchinario.

Note di implementazione

Devi implementare la seguente funzione:

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

- n: il numero di funghi raccolti da Andrew.
- Questa funzione viene chiamata esattamente una volta e deve restituire il numero di funghi di tipo A.

La funzione precedente può chiamare la seguente funzione:

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

- x: un array di lunghezza tra 2 e n (inclusi), contenente gli indici dei funghi da analizzare, nell'ordine di inserimento nella macchina.
- Gli elementi di x devono essere interi **distinti** da 0 a $n-1$ inclusi.
- Sia d la lunghezza dell'array x, la funzione restituisce il numero di indici distinti j, tali per cui $0 \leq j \leq d-2$ e i funghi $x[j]$ e $x[j+1]$ sono di specie diverse.
- Questa funzione può essere chiamata al massimo $20\,000$ volte.
- La somma delle lunghezze di x tra tutte le chiamate a `use_machine` non può superare $100\,000$.

Esempi

Esempio 1

Considera uno scenario in cui ci sono 3 funghi di tipo $[A, B, B]$, in ordine. La funzione `count_mushrooms` viene chiamata così:

```
count_mushrooms(3)
```

La funzione può chiamare `use_machine([0, 1, 2])` che restituisce 1. Può quindi chiamare `use_machine([2, 1])` che restituisce 0.

A questo punto ci sono abbastanza informazioni per concludere che c'è un solo fungo di tipo A, quindi la funzione `count_mushrooms` deve restituire 1.

Esempio 2

Considera un caso in cui ci sono 4 funghi dei seguenti tipi: $[A, B, A, A]$. La funzione `count_mushrooms` è chiamata nel seguente modo:

```
count_mushrooms(4)
```

La funzione può chiamare `use_machine([0, 2, 1, 3])` che restituisce 2.

Può quindi chiamare `use_machine([1, 2])` che restituisce 1.

A questo punto ci sono abbastanza informazioni per concludere che ci sono 3 funghi di tipo A, quindi la funzione `count_mushrooms` deve restituire 3.

Limitazioni

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

Assegnazione del punteggio

In ogni caso di test, se le chiamate a `use_machine` non rispettano i requisiti, oppure il valore restituito da `count_mushrooms` è errato, il punteggio sarà 0.

Altrimenti, sia Q il massimo numero di chiamate alla funzione `use_machine` tra tutti i casi di test, il punteggio è calcolato secondo la seguente tabella:

Condizione	Punteggio
$20\,000 < Q$	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

In alcuni casi di test il comportamento del grader è adattivo, per cui non c'è una sequenza di funghi fissata ma le risposte del grader possono dipendere dalle precendenti chiamate a `use_machine`. È comunque garantito che il grader risponda sempre in modo consistente.

Grader di esempio

Il grader di esempio legge l'input nel seguente formato:

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

dove $s[i] = 0$ indica che l'i-esimo fungo è di tipo A, mentre è di tipo B se $s[i] = 1$.

Il grader di esempio scrive l'output nel seguente formato:

- riga 1: il valore restituito da `count_mushrooms`.
- riga 2: il numero di chiamate a `use_machine`.

Nota che il grader di esempio **non** è adattivo.