

Poisonous nim (nim)

Stai giocando una partita di *Nim Avvelenato* contro Elia. Il Nim Avvelenato è giocato da due giocatori usando N pile di monete. Le pile sono numerate da 0 a $N - 1$ e l' i -esima pila contiene inizialmente A_i monete.

I giocatori si alternano nel fare mosse: una mossa consiste nello scegliere una pila non vuota e rimuovere da essa un numero positivo (cioè maggiore o uguale a 1) di monete.

Però, ogni volta che un giocatore rimuove monete da una pila, la pila diventa avvelenata e non può essere scelta dall'avversario nella sua prossima mossa. Nota che una pila rimane avvelenata solo per una mossa.

Il primo giocatore che non può fare alcuna mossa perde.

Tu inizi per primo, e vuoi scoprire quante diverse mosse iniziali ti garantiranno la vittoria supponendo che sia tu che Elia giochi in modo ottimale. Due mosse sono considerate diverse se rimuovono monete da pile diverse o il numero di monete rimosse è diverso.

Implementazione

Dovrai inviare un singolo file, con estensione `.cpp`, contenente la seguente funzione:

C++

```
long long play(int N, vector<int> A);
```

- La funzione verrà chiamata una volta durante l'esecuzione del tuo programma.
- L'intero N è il numero di pile di monete.
- L'array A contiene il numero di monete in ogni pila.
- La funzione deve restituire il numero di mosse iniziali che ti garantiscono la vittoria.

Grader di prova

Una versione semplificata del grader usato per la correzione è disponibile nella directory relativa a questo problema. Puoi usarla per testare le tue soluzioni in locale. Il grader legge i dati di input da `stdin`, chiama la funzione che devi implementare e scrive su `stdout` nel seguente formato.

Il file di input è composto da 2 righe, contenenti:

- Riga 1: l'intero N .
- Riga 2: N interi $A_0 \dots A_{N-1}$.

Il file di output è composto da una riga, contenente il valore restituito dalla funzione `play`.

Assunzioni

- $2 \leq N \leq 1\,000\,000$.
- $1 \leq A_i \leq 1\,000\,000\,000$, per ogni i da 0 a $N - 1$.

Assegnazione del punteggio

- **Subtask 0** [0 punti]: Casi d'esempio.
- **Subtask 1** [11 punti]: $N = 2$.
- **Subtask 2** [8 punti]: $N \leq 4$, $A_i \leq 50$.
- **Subtask 3** [19 punti]: $N \leq 100$, $A_i \leq 100$.

- Subtask 4 [12 punti]: $N \leq 100$, $A_i \leq 10\,000$.
- Subtask 5 [17 punti]: $N \leq 1000$, $A_i \leq 10\,000$.
- Subtask 6 [11 punti]: $N \leq 1000$.
- Subtask 7 [9 punti]: $N \leq 100\,000$.
- Subtask 8 [13 punti]: Nessuna limitazione aggiuntiva.

Esempi di input/output

stdin	stdout
3 1 5 1	4
3 4 6 5	6

Spiegazione

Nel **primo caso d'esempio** ci sono 4 mosse iniziali che ti garantiscono la vittoria (nota che le pile sono indicizzate a partire da 0):

- rimuovere 1 moneta dalla pila con indice 1.
- rimuovere 2 monete dalla pila con indice 1.
- rimuovere 3 monete dalla pila con indice 1.
- rimuovere 5 monete dalla pila con indice 1.

Nel **secondo caso d'esempio** ci sono 6 mosse iniziali che ti garantiscono la vittoria.