

## Poisonous nim (nim)

Tu joues à une partie de *Nim Empoisonné* contre Binna. Le Nim Empoisonné se joue à deux avec  $N$  piles de pièces. Les piles sont numérotées de 0 à  $N - 1$  et la  $i$ -ème pile contient initialement  $A_i$  pièces.

Les joueurs jouent à tour de rôle : un coup consiste à choisir une pile non vide et à en retirer un certain nombre de pièces, entre une et le nombre actuel de pièces dans la pile.

Cependant, chaque fois qu'un joueur retire des pièces d'une pile, celle-ci devient empoisonnée et ne peut pas être choisie par l'adversaire à son prochain tour. Note qu'une pile ne reste empoisonnée que pour un seul tour.

Le premier joueur qui ne peut plus faire de coup perd.

C'est à toi de commencer, et tu veux savoir combien de premiers coups différents te garantiront la victoire, en supposant que toi et Binna jouez de manière optimale. Deux coups sont considérés comme différents s'ils retirent des pièces de piles différentes ou si le nombre de pièces retirées est différent.

## Implémentation

Tu devras soumettre un unique fichier, avec une extension `.cpp`, contenant la fonction suivante.

C++	<code>long long play(int N, vector&lt;int&gt; A);</code>
-----	--

- La fonction sera appelée une seule fois pendant l'exécution de ton programme.
- L'entier  $N$  est le nombre de piles de pièces.
- Le tableau  $A$  contient le nombre de pièces dans chaque pile.
- La fonction doit retourner le nombre de premiers coups qui te garantissent la victoire.

## Évaluateur

Un évaluateur d'exemple est fourni. Il lit depuis `stdin` et écrit dans `stdout` selon le format suivant :

Le fichier d'entrée est composé de 2 lignes, contenant :

- Ligne 1 :  $N$ .
- Ligne 2 :  $N$  entiers  $A_0 \dots A_{N-1}$ .

Le fichier de sortie est composé d'une seule ligne, contenant la valeur retournée par la fonction `play`.

## Contraintes

- $2 \leq N \leq 1\,000\,000$
- $1 \leq A_i \leq 1\,000\,000\,000$ , pour chaque  $i$  de 0 à  $N - 1$ .

## Score

- **Sous-tâche 0 [ 0 points ]**: Exemples
- **Sous-tâche 1 [11 points]**:  $N = 2$
- **Sous-tâche 2 [ 8 points ]**:  $N \leq 4, A_i \leq 50$
- **Sous-tâche 3 [19 points]**:  $N \leq 100, A_i \leq 100$
- **Sous-tâche 4 [12 points]**:  $N \leq 100, A_i \leq 10\,000$
- **Sous-tâche 5 [17 points]**:  $N \leq 1000, A_i \leq 10\,000$
- **Sous-tâche 6 [11 points]**:  $N \leq 1000$

- **Sous-tâche 7 [ 9 points]**:  $N \leq 100\,000$
- **Sous-tâche 8 [13 points]**: Aucune contrainte supplémentaire.

## Exemples

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

## Explication

Dans le **premier exemple**, il y a 4 premiers coups qui te garantissent la victoire (note que les piles sont indexées à partir de 0) :

- retirer 1 pièce de la pile d'indice 1.
- retirer 2 pièces de la pile d'indice 1.
- retirer 3 pièces de la pile d'indice 1.
- retirer 5 pièces de la pile d'indice 1.

Dans le **deuxième exemple**, il y a 6 premiers coups qui te garantissent la victoire.