

**Summen der Form $a + b = c$**

Task ID: 3sum	Time limit: 0.4 sec	Memory limit: 64 MB
---------------	---------------------	---------------------

Eingabe

Dies ist die Aufgabe 2.c) der Serie 3, indem du zusätzlich noch die Anzahl Lösungen zählen sollst.

Die erste Zeile enthält N , $3 \leq N \leq 10^4$.

Die zweite Zeile enthält a_1, \dots, a_N , wobei $1 \leq a_i \leq 10^9$ für alle $1 \leq i \leq N$.

Beachte, dass diese Zahlen a_i nicht sortiert sein müssen und die selbe Zahl mehrfach vorkommen kann.

Limits

Die Eingabe besteht aus 4 Testgruppen, jede gibt 25 Punkte.

- Für 25 Punkte gilt $N \leq 100$.
- Für 50 Punkte gilt $N \leq 1000$ (erlaubt $\mathcal{O}(N^3)$ Ansätze).
- Für 100 Punkte gilt $N \leq 5'000$ (erlaubt $\mathcal{O}(N^2)$ Ansätze).
- Für 1'000'000 CHF sind Ansätze gesucht, die in $\mathcal{O}(N^{2-c})$ sind, für irgendein $c > 0$;-)

Ausgabe

Finde heraus wie viele *Kombinationen* es gibt, von der Form $a_i + a_j = a_k$ für $1 \leq i < j \leq N$ und $1 \leq k \leq N$ und gib diese Anzahl aus.

Beispiele

input	output			
<table border="1"><tr><td>6</td></tr><tr><td>5 1 2 3 7 3</td></tr></table>	6	5 1 2 3 7 3	<table border="1"><tr><td>5</td></tr></table>	5
6				
5 1 2 3 7 3				
5				

Erklärung: Folgende Möglichkeiten sollen gezählt werden:

- $1 + 2 = 3$ (mit der ersten 3)
- $1 + 2 = 3$ (mit der zweiten 3)
- $2 + 3 = 5$ (mit der ersten 3)
- $2 + 3 = 5$ (mit der zweiten 3)
- $5 + 2 = 7$



input

```
10
2 2 1 1 1 1 2 2 3 3
```

output

```
56
```

Hinweise

- Der Grader für diese Zusatzaufgaben akzeptiert Einsendungen in Java, C, C++, Python und Pascal.
- Für Einsendungen in Java muss die Klasse *submission* heissen, vgl. die Templates auf der Webseite.
- Da die Eingabe sehr gross werden kann, beachtete folgende Hinweise:
 - Für Java: verwende `BufferedReader` statt `Scanner` (siehe Template).
 - Für C/C++: Benutze entweder C/C++ `printf/scanf` oder verbessere die Performance von C++ streams indem du `std::ios_base::sync_with_stdio(false)`; am Anfang von *main* inkludierst.