Zadanie: ZOL
Żołnierze

Etap III. Plik źródłowy `zol.*` *10.05.2008*

Dostępna pamięć: 32 MB.

Na porannym apelu w koszarach wszyscy przebywający tam żołnierze muszą ustawić się w szeregu. Nie mogą jednak stanąć w dowolnej kolejności, tylko od najwyższego do najniższego. Najwyższy może przy tym stać zarówno z lewej, jak i z prawej strony. Pomóż im wyznaczyć liczbę sposobów, na jakie mogą ustawić się poprawnie.

Dwa ustawienia żołnierzy uważamy za identyczne wtedy i tylko wtedy, gdy każdy żołnierz w obu ustawieniach ma tego samego sąsiada po lewej stronie (lub w obu nie ma go wcale) oraz w obu ustawieniach ma tego samego sąsiada po prawej stronie (lub w obu nie ma go wcale).

Zadanie

Napisz program, który:

- wczyta ze standardowego wejścia opis wszystkich przebywających w koszarach żołnierzy,
- wyznaczy liczbę ustawień żołnierzy w jednym rzędzie w kolejności od najwyższego do najniższego,
- wypisze cztery ostatnie cyfry dziesiętne wyniku na standardowe wyjście.

Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba całkowita n ($1 \le n \le 200\,000$), oznaczająca liczbę żołnierzy w koszarach. Drugi wiersz wejścia zawiera n liczb całkowitych w_i ($1 \le w_i \le 10^9$), pooddzielanych pojedynczymi odstępami i oznaczających wysokości kolejnych żołnierzy.

Wyjście

Twój program powinien wypisać w pierwszym i jedynym wierszu wyjścia cztery ostatnie cyfry dziesiętne liczby ustawień żołnierzy w jednym rzędzie w kolejności posortowanej względem ich wysokości. Jeżeli wynik jest mniejszy niż 1 000, to należy wypisać wszystkie jego cyfry.

Przykład

Dla danych wejściowych:

```
7
2 3 1 4 4 5 2
```

poprawnym wynikiem jest:

```
8
```

Uwaga: Możesz założyć, że w 50% testów jest spełniony warunek $n \le 5\,000$.

Komentarz do przykładu

Oto wszystkie poprawne ustawienia żołnierzy z przykładu (w nawiasach znajdują się wysokości żołnierzy, a poza nimi — ich numery, zgodne z kolejnością z wejścia):

3 (1), 1 (2), 7 (2), 2 (3), 4 (4), 5 (4), 6 (5)
3 (1), 7 (2), 1 (2), 2 (3), 4 (4), 5 (4), 6 (5)
3 (1), 1 (2), 7 (2), 2 (3), 5 (4), 4 (4), 6 (5)
3 (1), 7 (2), 1 (2), 2 (3), 5 (4), 4 (4), 6 (5)
6 (5), 4 (4), 5 (4), 2 (3), 1 (2), 7 (2), 3 (1)
6 (5), 5 (4), 4 (4), 2 (3), 1 (2), 7 (2), 3 (1)
6 (5), 4 (4), 5 (4), 2 (3), 7 (2), 1 (2), 3 (1)
6 (5), 5 (4), 4 (4), 2 (3), 7 (2), 1 (2), 3 (1)