

Sortowanie bąbelkowe nie jest oczywiście najszybszym algorytmem sortowania. Nie zawsze jednak zależy nam na wydajności, czasami chcemy prostego algorytmu.

Algorytm polega na wielokrotnym wykonaniu przelotu (aż do posortowania ciągu). Każdy przelot polega na porównywaniu kolejnych par sąsiednich elementów (od lewej do prawej). Jeśli element po lewej jest większy niż po prawej, algorytm dokonuje zamiany tych elementów miejscami. W przeciwnym przypadku algorytm nie wykonuje na tych elementach żadnej akcji.

Napisz program, który: wczyta ciąg liczb, posortuje go z użyciem sortowania bąbelkowego i wypisze wynik na standardowe wyjście.

WEJŚCIE

W pierwszym wierszu wejścia znajduje się jedna liczba naturalna N , określająca długość ciągu. W drugim (ostatnim) wierszu znajduje się ciąg N liczb całkowitych T_i , pooddzielanych pojedynczymi odstępami. Są to liczby ciągu, który należy posortować.

WYJŚCIE

Twój program powinien wypisać na wyjście dokonane zamiany w kolejności ich wykonywania, po jednej w wierszu. Opis każdej zamiany powinien się składać z dwóch liczb całkowitych, oddzielonych pojedynczym odstępem — zamienianych elementów.

Ostatni wiersz wyjścia powinien zawierać posortowany ciąg. Elementy ciągu powinny być pooddzielane pojedynczymi odstępami.

OGRANICZENIA

$$1 \leq N \leq 1\,000, 0 \leq T_i \leq 10^9.$$

PRZYKŁAD

Wejście	Wyjście
6	3 1
3 1 8 2 8 1	8 2
	8 1
	3 2
	8 1
	3 1
	2 1
	1 1 2 3 8 8