

Odwracane sortowanie (odwracane-sort)

Memory limit: 32 MB

Time limit: 4.00 s

Bajtocki Urząd Sortujący zakupił właśnie w drodze przetargu ultranowoczesną maszynę sortującą. Dokładniej, to nie tyle sortującą co odwracającą spójne podciągi zadanego ciągu w czasie stałym. Na przykład jedną operacją z ciągu 1 2 3 4 5 6 można uzyskać ciąg: 1 5 4 3 2 6. Fajne, prawda?

Urzednicy puścili wodzę fantazji i chcieliby zastosować maszynę do posortowania permutacji zbioru liczb $\{1, 2, \dots, N\}$. Wiadomo bowiem, że zawsze wystarczy liniowo wiele odwróceń spójnych podciągów, aby posortować dowolny ciąg – z użyciem ultramaszyny można sortować w czasie liniowym! No prawie. Trzeba jeszcze umieć powiedzieć maszynie jakie kawałki ciągu ma odwracać, co może nie być już takie łatwe.

Maszyna przyjmuje jako dane ciąg do posortowania długości N oraz ciąg N operacji. Jeśli i -ta liczba ciągu operacji jest równa B_i to w i -tej operacji nastąpi odwrócenie spójnego podciągu od i -tego do B_i -tego elementu włącznie. Elementy ciągu numerujemy kolejnymi liczbami naturalnymi od 1 do N włącznie.

Napisz program, który: wczyta permutację do posortowania, wyznaczy ciąg operacji dla maszyny sortującej i wypisze wynik na standardowe wyjście.

Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba naturalna N , określająca długość permutacji do posortowania. W drugim (i ostatnim) wierszu wejścia znajduje się ciąg N liczb naturalnych A_i , pooddzielanych pojedynczymi odstępami – elementy ciągu.

Wyjście

Twój program powinien wypisać na wyjście N liczb całkowitych, pooddzielanych pojedynczymi odstępami. i -ta liczba (B_i) powinna określać, że i -tą operacją sortowania powinno być odwrócenie spójnego podciągu od i -tego elementu do B_i -tego (włącznie).

Ograniczenia

$1 \leq N \leq 300\,000$.

Przykład

Input

```
7
5 2 6 7 4 1 3
```

Output

```
6 5 7 5 5 7 7
```