

(1) 入力例

入力は、1 行目に整列対象要素数 n ($1 \leq n \leq 100000$ 程度を想定) を 10 進整数で与え、続く n 行に符号付き 32 ビット範囲内の整数 a_i を 1 行 1 個ずつ標準入力から与える形式であり、プログラムでは動的確保した配列 x に順次格納した後、選択ソートによる整列処理を行う。ここでは空白や追加の区切り文字を許さず改行だけを区切りとし、 n が負値である場合は未定義入力として動作保証外とするが、採点環境では必ず正の整数が与えられる前提である。

(2) 出力例

出力は、昇順へ整列後の配列要素を先頭から末尾まで半角空白で区切って 1 行に列挙し、最後の要素の直後には区切り文字を置かず直ちに改行コードを出力し、続く 2 行目に選択ソート中で実際に実施した要素間比較回数 b を 10 進整数で出力し改行によって終了する仕様である。比較回数は内部でインクリメントされるたびに累積されるため入力系列に応じて 0 から $n(n-1)/2$ まで変化し、両行とも数値と区切りの空白および改行のみが含まれ不要な文字列が混入しない設計となっている。

(3) 入力に対する出力結果の妥当性の説明

本プログラムは i を 0 から $n-2$ まで進めつつ未整列部分の最小値を探索し、 min が更新されるたびに swap を介して先頭 i 番目の要素と交換する典型的な選択ソートを実装しており、内部で j を走査するたびに比較回数 b を 1 ずつ加算することで総比較回数を正確に計測する。実行例として逆順入力 $n=5$, $5\ 4\ 3\ 2\ 1$ に対しては理論値 10 の比較を経て $1\ 2\ 3\ 4\ 5$ が得られ、結果は教科書記載の計算量 $O(n^2)$ と一致しメモリリークも free により回避されていることから、課題要求である正確な整列処理と比較回数の算出をも満たしていると判断できる。