

(1) 入力例

入力は、最初の行に配列の要素数 n ($1 \leq n \leq 100000$ 程度まで想定) を十進整数で与え、続く n 行に並び替え前の整数 a_i を 1 行 1 個ずつ入力する形式であり、例えば $n=5$ のとき「5」「5」「4」「3」「2」「1」の 6 行が与えられ、プログラムはこれらを動的確保した配列 x に格納し、整数は符号付き 32 ビット範囲内とし空白や余分な改行を許容せず標準入力から逐次読み取る。入力値の順序はそのままアルゴリズムが保持し、途中で検査用のトークンや区切り文字を置かないことを前提とする。また、 n が 0 以下の場合仕様外とし実行を保証しないが、授業の採点環境では必ず 1 以上が与えられる。読み取りは `scanf` を用いているため、全ての数値は半角数字と改行コードのみで構成されることがあり、末尾の改行が欠けても動作するが推奨されない。

(2) 出力例

出力は、挿入ソートにより昇順に整列された配列要素を先頭から末尾まで半角空白 1 文字で区切って列挙し行末にも空白を残した状態で改行し、続く 2 行目にソート過程で実際に右隣へシフトした要素の総回数 `counter` を十進整数で表示して改行して終了する仕様であり、例えば前項の入力例に対しては「1 2 3 4 5」に続く改行と「10」に続く改行の計 2 行が出力される。ここで `counter` は逆転数に一致するため入力順序によって 0 から $n(n-1)/2$ まで変化し、いかなる入力に対しても数値、空白、改行のみから成るため自動評価時に余分な文字列が混入しない。さらに、`printf` 関数を用い書式指定子「%d」で各整数を符号付き十進表記のまま出力するため、不要な先頭ゼロや桁区切り記号は挿入されない。

(3) 入力に対する出力結果の妥当性の説明

プログラム内部では、添字 i を 1 から $n-1$ まで増分しつつ未整列領域の先頭 $a[i]$ を一時変数 `tmp` に退避し、左隣の値と比較しながら条件 $a[j-1] > tmp$ を満たす間は要素を右へ 1 つずつシフトすることで挿入位置を空ける挿入ソートを実装しており、このシフト操作を行うたびに計数器 `counter` をインクリメントして逆転解消回数を正確に蓄積するため、アルゴリズムの安定性と $O(n^2)$ の最悪計算量という理論値を忠実に反映する。実行例では最も逆転が多い降順入力 5 4 3 2 1 に対し理論上の逆転数 10 と整列済み配列 1 2 3 4 5 が出力され、標準的なテストケースと照合しても値と順序が一致することから、入力仕様を満たす全ての場合に正しく機能しメモリリークも `free(x)` により解放される点で、課題の要件を完全に満たしていると結論付けられる。