

(1) 入力例

入力は標準入力から与え、最初に行列の個数 n を整数で 1 つ入力し ($1 \leq n \leq 100$ で範囲外はエラー終了), 続いて $i=1$ から n まで各行列 A_i の行数 $r[i]$ と列数 $c[i]$ を 2 整数として空白区切りで入力する形式であり, 入力全体は改行区切りで与えてよくプログラムは途中で「行列の個数 n を入力してください:」「行列 $A_1 (r[1] \ c[1]):$ 」のような案内を表示するため, 例えば「3 10 30 30 5 5 60」と入力し, 連鎖積が定義できる条件 $c[i]=r[i+1]$ ($i=1 \cdots n-1$) と各次元が正である条件を満たすように準備する, また $n=1$ の場合は行列 1 個のみを入力し演算回数 0 となることを想定し, $r[i]$ と $c[i]$ は `int` で扱うため極端に大きい値は避け, 数値以外の文字列や不足入力を含めないよう注意する, さらに本実装では $c[i]=r[i+1]$ の整合性検査を行わないため, 入力段階で連鎖可能性を確認することが必須である.

(2) 出力例

出力は, 入力された行列列 $A_1 \cdots A_n$ をどの順に括弧付けして掛け合わせるかを変えたときのスカラー乗算回数 (積の計算で行う乗算回数) の最小値を表す $M[1][n]$ を 1 行で表示するものであり, 表示文字列は固定の接頭辞「最小の演算回数:」に続けて 10 進整数を出力し最後に改行を付ける仕様であるため, 例えば入力例「3 10 30 30 5 5 60」では最小回数が 4500 となり画面には「最小の演算回数: 4500」と表示され, $n=1$ のときは「最小の演算回数: 0」となる, なお途中で入力を促す文言を表示するが最終結果として採点対象となるのは上記 1 行であり, 出力値はアルゴリズムの計算結果を `int` へ格納して表示するため通常は 32bit 符号付き整数範囲に収まるよう入力次元を設定することを前提とする, 出力先は標準出力であり余分な空白や追加行は付加しないことを想定する, したがって自動判定では文字列と数値が完全一致するよう確認する.

(3) 入力に対する出力結果の妥当性の説明

添付資料が未提供のためここではソースコードから読み取れる範囲で考察すると, 本プログラムは動的計画法により区間 $[i, j]$ の最小コスト $M[i][j]$ を $w=1$ から $n-1$ へ増やしながら計算し, 初期条件 $M[i][i]=0$ とし未計算領域は `INT_MAX` で擬似的に $+\infty$ を表現した上で, 分割点 k ごとに $q=M[i][k]+M[k+1][j]+r[i] \times c[k] \times c[j]$ を評価して最小値へ更新するため, 入力例「3 10 30 30 5 5 60」では $(A_1 A_2) A_3$ の 4500 と $A_1 (A_2 A_3)$ の 27000 を比較して 4500 を出力し要件の「最小の演算回数」を満たすことが確認でき, 計算量は三重ループで $O(n^3)$ かつ配列 M の $O(n^2)$ であり $n \leq 100$ なら実行可能である一方, 次元不整合の検査が無い点と q を `long long` で計算しても $M \rightarrow \text{int}$ で代入する点から極端な入力ではオーバーフローや不正解につながり得る.