

(1) 入力例

本プログラムの入力は、整列済み整数列に対して二分探索を行うためのデータであり、最初の行に列の要素数 n ($1 \leq n \leq 10000$) を 10 進整数で与え、続く n 行に昇順に並んだ整数 $v_1 \dots v_n$ を 1 行 1 個ずつ記述し、最後に探索対象の整数 k を別行で入力する形式であり、各値は `int` 型範囲内で空白文字は改行のみを用い、例えば要素数 6 の列 $\{1, 2, 3, 4, 5, 6\}$ から値 2 を探索する場合入力は「6」「1」「2」「3」「4」「5」「6」「2」の 8 行となり、プログラムは `scanf` を用いて順に読み取って配列 `d` に格納すると同時に n を参照して左端 `left=0` 右端 `right=n-1` を初期化し後続の探索処理に引き渡す仕様である。

(2) 出力例

出力は、読み取り途中でユーザに入力を促すプロンプトとして「`n?:`」「`k?:`」が改行付きで表示され、探索が成功した場合には「**Found:** 添字 値」の書式で配列中の 0 から始まる添字と探索値を出力し、失敗した場合には「**Not Found:** 値」を出力して処理を終了し、プロンプトは評価対象外であるが提出規約には同封可能とされ、前述の入力例に対してはプロンプトの後に「**Found: 1 2**」が表示され添字 1 が配列中での 2 の位置であることを示し、出力末尾に余分な空白や改行が入らないよう留意する必要がある。

(3) 入力に対する出力結果の妥当性の説明

二分探索アルゴリズムは整列済み配列に対し比較回数を最大 $\lceil \log_2 n \rceil$ に抑えて目的値の有無を判定でき、本実装では `left ≤ right` をループ条件とすることで配列外参照を防止しつつ探索範囲を半減し条件分岐ごとに `mid` を再計算するため無限ループを生じず、入力例では比較順に `mid=2` (値 3) `mid=0` (値 1) `mid=1` (値 2) と 3 回で目的値に一致して添字 1 が返され `main` 関数がこれを検査して「**Found**」を出力しており、理論的結果と一致していることからプログラムは問題要件を満たしていると判断できる。