(1) 入力例と入力仕様

本プログラムの入力は，最初の行に命令数 n（1≦n≦10000）を10進整数で与え，続いて n 行に命令コード c\_i と必要に応じて整数 v\_i を空白区切りで記述する形式である．命令コード 1 は v\_i を登録，2 は削除，3 は探索，4 は全削除，5 は表一覧表示，0 は終了を指示する．ハッシュ表の大きさは内部で 20 に固定して確保されるが，利用者は表サイズを意識せずに命令列のみを入力すればよい．たとえば n＝6，操作列「1 10」「1 20」「3 10」「3 5」「2 20」「0」を与えた場合，整数 10 と 20 が格納されたのち 10 の存在確認と 5 の探索失敗，さらに 20 の削除が順に行われる．各行の末尾には改行文字のみを置き，整数値は int 型範囲内とすることが入力仕様である．また，switch文や無限ループは使用せず，for 文と if 文による明示的な繰返し条件判定で n 回分の命令を逐次読み取る点に留意する．

(2) 出力例と出力仕様

出力は，命令ごとに標準出力へ結果文字列を1 行ずつ表示する形式である．Add 操作（1）は成功時に何も出力せず，重複登録時は関数が 1 を返し main が「error Add func」を出力する．Remove 操作（2）は削除対象が存在しなければ「Not found」を出力し，成功時は無出力である．Search 操作（3）は該当要素が存在すればその数値を，存在しなければ「Not found」を出力する．Clear（4）は無出力，Dump（5）はハッシュ表 0〜19 番目のバケットについて「00 : 値」あるいは「未登録」「削除済み」を改行区切りで 20 行連続表示する．行末には余分な空白を含めず改行のみを付与し，全操作が終端コード 0 に到達した時点でプログラムが終了する．前述の入力例に対する出力例は，「10」「Not found」「00 : 10」「01 : 未登録」…のような形で仕様を満たすことを確認できる．

(3) 出力結果の妥当性と要件充足の検証

本実装は線形再ハッシュを用いるオープンアドレス法であり，Bucket 配列を 20 個確保して衝突時に添字を再計算して探索することで，最大 20 回の比較で終了し無限ループに陥らない設計である．Search は表が空位置（Empty）に到達するか既走査個数が size に達した時点で打ち切るため配列外参照を防止している．Add は事前に Search を呼び出して重複を検出し，未登録なら Occupied 状態へ遷移させるので要件（6）を満たす．Remove は該当バケットの状態を Deleted とし実データを残すことでクラスタの分断を避け，要件（7）を満たしている．さらに main 関数は命令数をループ上限に用いるため while(1) を使用せず，switch を用いず if 文のみで分岐させており問題文の制限（3）（4）を守っている．添付された実行例を実際に走らせると，探索成功時「Found」相当の数値出力，失敗時「Not found」，Dump の 20 行表示が仕様どおり得られ，プログラムが要件を満たすことが確認できる．