

## Hash\_open\_char.c

```
// ハッシュ表に入力値を格納
// 入力値: *table ハッシュ表
//      size ハッシュ表の大きさ
//      cons 除数
//      inv 入力値
// 戻り値: バケット番号 or -1(格納済)
int store(CELL *table, unsigned int size, unsigned int cons, char *inv) {
    unsigned int adr, // ハッシュ表の番地(添字)
                count = 1; // 探査回数

    adr = hash_str(inv); // 文字列を 10 進数に変換
    adr = hash(adr, cons); // hash 値を計算
    TABLE[adr]++; // 当該番地の衝突回数を加算
    while (count <= size) {
        if (table[adr].key == INTV) { // 当該番地が空いている
            table[adr].key = strdup(inv); // 入力値を格納
            return adr;
        }
        else if (table[adr].key != INTV) { // 当該番地は空いていない
            if (strcmp(table[adr].key, inv) == 0) // 入力値が既格納
                return FLS; // 終了
            adr = prob_lin(size, CONSL, adr); // 次の空き番地を線形探査法で計算
            count++; // 探査回数を加算
        }
    }
    return FLS;
}
```