

```

// 入力値： key      キー
//          mod      除数(素数)
// 戻り値： hv      バケット番号
int hash(int key, int mod) {          // 剰余法
    int hv ;                          // 剰余
    hv = key % mod ;                  // 剰余を hash 値とする
    Table[hv]++;
    return hv ;
}

// 入力値： *table    hash 表
//          size     hash 表の大きさ
//          cons     除数
//          inv      入力値
// 戻り値：          バケット番号、-1(既格納)
int store(CELL *table, int size, int cons, int inv) {
    unsigned int adr ,                // hash 表の番地(添え字)
                count = 1 ;          // 探査回数

    adr = hash(inv, cons) ;           // hash 値を計算
    while(count <= size) {
        if (table[adr].key == INTV) { // adrsi 番地は空いている
            table[adr].key = inv ;    // adrs 番地に格納
            Frq_prob += count;
            return adr ;
        }
        else if (table[adr].key != INTV) { // adrs 番地は空いていない
            if (table[adr].key == inv) // 入力値は既格納
                return FLS ;
            adr = prob_lin(size, CONSL, adr) ; // 次の空き番地(線形探査法)
            ++count ;
        }
    }
    return FLS ;
}

```