Hash_open_char.c

```
/*
  ハッシュ表(開放番地法)の操作
  キー      :  文字列

  store     :  キーをハッシュ化し配列に格納する
  lookup    :  キーを配列中から探す
  delete    :  配列中からキーを削除する
*/



// 入力値 :  table   ハッシュ表の先頭番地
//        :  size    ハッシュ表の大きさ(番地空間)
//        :  cons    ハッシュ関数の除数
//        :  inv     格納要素
// 戻り値 :          格納(0 <), 非格納(-1), 異常(-2)
int store(CELL *table, unsigned int size, unsigned int cons, char *
inv) {
  unsigned int adr ,
    count = 1 ;
  char error[80] = "ハッシュ表が溢れました" ;

  adr = hash_str(inv) ;                  // 文字列を10進整数値化
  adr = hash(adr, cons) ;                 // hash値を計算
  Table[adr]++;
  while(count < size) {
    if (table[adr].key == INTV) { // i番地は空いている
      table[adr].key = strdup(inv) ;       // i番地に格納
      Frq_prob += count;
      return adr ;
    }
    else if (table[adr].key != INTV) {     // adr番地は空いていない
      if (strcmp(table[adr].key, inv) == 0)      // 入力値は既格納
    return FLS ;
      adr = prob_lin(size, CONSL, adr) ;            // 次の番地
      ++count ;
    }
```

```
  }
  return FLS ;                                    // ハッシュ表が溢れた
}
```