```c
/*
 * This code is from "Algorithms in C, Third Edition,"
 * by Robert Sedgewick, Addison Wesley Longman, 1998.
 * (Mescla de prog12.5.c e prog14.3.c)
 */
#include "Item.h"
#include "ST.h"

typedef struct STnode *link;
struct STnode { Item item; link next; };

static link *heads, z;
static int N, M;

static link NEW(Item item, link next)
{
    link x = malloc(sizeof *x);
    x->item = item;
    x->next = next;
    return x;
}

int STcount()
{
    return N;
}

/* prog14.1 */

int hash(char *v, int M)
  { int h = 0, a = 127; /* a = 128; */
    for (; *v != '\0'; v++)
      h = (a*h + *v) % M;
    return h;
  }

#define hash(v,M) (hash(v,M))

void STinit(int max)
  { int i;
    N = 0;
    M = 97;
    /* M = 96; */
    heads = malloc(M*sizeof(link));
    z = NEW(NULLitem, NULL);
    for (i = 0; i < M; i++) heads[i] = z;
  }

Item searchR(link t, Key v)
  {
    if (t == z) return NULLitem;
    if (eq(key(t->item), v)) return t->item;
    return searchR(t->next, v);
  }

Item STsearch(Key v)
  { return searchR(heads[hash(v, M)], v); }

void STinsert(Item item)
  { int i = hash(key(item), M);
    heads[i] = NEW(item, heads[i]); N++; }

void STdump(void (*visit)(Item))
{
```

```
    int i;
    link t;

    for (i=0; i<M; i++) {
      if (heads[i] != z)
        for (t = heads[i]; t != z; t = t->next)
          visit(t->item);
    }
}

void STshow_histogram()
{
    int i, j, no_per_list, max_list = -1;
    link t;

    for (i=0; i<M; i++) {
      printf("\n+ list %d: ", i);
      for (no_per_list=0, t = heads[i]; t != z; no_per_list++, t = t->next);
      printf("[%d] ", no_per_list);
      j = no_per_list;
      while (j-- > 0)
        printf("*");
      if (no_per_list>max_list) max_list=no_per_list;
    }

    printf("\nMax no. of items in the lists: %d\n", max_list);
    printf("Average no. of items in the lists: %g\n\n", (double)N/M);
}
```