```
        Start: 1 2 3
    1: move disk 1:   2 3
    2: move disk 2:    3     2
    3: move disk 1:    3     1 2
    4: move disk 3:         1 2     3
    5: move disk1:          2     3
    6: move disk 2:              23
    7: move disk 1:             1  23
```

To save space, the disks on each tower are written from left to right instead of vert ic ally. From this output, we can tell at a glance that a larger disk is never placed on a smaller one.

According to legend, each day one of the monks in the monastery that attends the three towers of Hanoi moves one of the disks. At the beginning of time, tower A started with 64 disks. When the last disk gets moved, according to legend, the world will come to an end.

Summary

Functions are the most general structuring concept in C. They should be used to implement "top-down" problem solving- namely, breaking up a problem into smaller and smaller subproblems until each piece is readily expressed in code.

2 A return statement terminates the execution of a function and passes control back to the calling environment. If the return statement contains an expression, then the value of the expression is passed back to the calling environment as well.

3 A function prototype tells the compiler the number and type of arguments that are to be passed to the function. The general form of a function prototype is

type function_name (*parameter type list*) ;

The parameter type list is typically a comma-separated list of types, with optional identifiers. If a function takes no arguments, then the keyword void is used. If the function returns no value, then the function type is void.

4 Arguments to functions are passed by value. They must be type compatible with the corresponding parameter types given by the function definition or the function prototype.

5 The storage class of a function is always extern, and its type specifier is int unless otherwise explicitly declared. The return statement must return a value compatible with the function type.

6 The principal storage class is automatic. Automatic variables appear and disappear with block entry and exit. They can be hidden when an inner block redeclares an outer block identifier.

7 Scope rules are the visibility constraints associated with identifiers. For example, if in a file we

```
have static void f(void)
{
    .....
}
static int a, b, c;

.....
```

then f() will be known throughout this file but in no other, and a, b, and c will be known only in this file and only below the place where they are declared.

8 The external storage class is the default class for all functions and all variables declared outside of functions. These identifiers may be used throughout the program. Such identifiers can be hidden by redeclaration, but their values cannot be destroyed.

9 The keyword extern is used to tell the compiler to "look for it elsewhere, either in this file or in some other file."

10 The storage class register can be used to try to improve execution speed. It is semantically equivalent to automatic.

11 The storage class static is used to preserve exit values of variables. It is also used to restrict the scope of external identifiers. This latter use enhances modularization and program security by providing a form of privacy to functions and variables.

12 External and static variables that are not explicitly initialized by the programmer are initialized to zero by the system.

13 A function is said to be recursive if it calls itself, either directly or indirectly. In C, all functions can be used recursively.