

# 实用 C语言软件 开发工具

李向荣 张宇 编著



清华大学出版社

# 实用 C 语言软件开发工具

李向荣 张 宇 编著

清华大学出版社

(京)新登字 158 号

### 内 容 简 介

本书精选了 200 多个实用 C 语言函数,功能覆盖了汉字的输入输出、外部设备控制、人机交互式界面、数据库、统计图表等多方面。详细介绍了各函数的基本原理、数据结构、功能、语法及使用说明,大多数函数还配有示范实例。

本书分为带盘和不带盘两种,盘中包含:函数的库文件、头文件和例子函数。读者可直接调用搭制自己的应用程序。

本书兼有示范性和实用性,目的是帮助读者学会编制具有漂亮的图形交互界面和良好数据结构的汉化软件。适用对象包括高等院校教师、从事 C 语言软件开发的程序员以及有志于深造的初学者。

本书配套软盘,清华大学出版社软件部有售。每套软盘售价(含邮资)43 元。

### 图书在版编目(CIP)数据

实用 C 语言软件开发工具/李向荣,张宇编著. —北京:清华大学出版社,1995

ISBN 7-302-01826-X

I. 实… II. 李… ② 张 III. C 语言-程序设计-软件开发 W. TP312C

中国版本图书馆 CIP 数据核字(95)第 05074 号

出版者:清华大学出版社(北京清华大学校内,邮编 100084)

印刷者:清华大学印刷厂

发行者:新华书店北京科技发行所

开 本:787×1092 1/16 印张:16 字数:368 千字

版 次:1995 年 7 月第 1 版,1996 年 3 月第 2 次印刷

书 号:ISBN 7-302-01826-X/TP·818

印 数:1501—5500

定 价:24.00 元

# 前 言

本书并不从头教你如何使用 C 语言,它是专为那些刚刚学会 C 语言,但不知如何着手开发实用软件,或者想开发软件却苦于手头没有很好开发工具的人们编写的。

C 语言是当今世界最广泛流行的软件开发语言,其简洁灵活的句法,丰富实用的函数库是各种软件开发的有力工具。但是,如果你直接使用 C 语言来进行哪怕是最一般的实用软件开发,也会发现:由于 C 语言所提供的函数库的功能过于基本,开发的过程仍是相当费时费力的。本书就是为了解决上述问题而编写的。

本书的内容是 C 语言扩展函数库及其编程技巧。这里的“扩展函数库”,是指本书中不包括在 Borland C++ 或 Turbo C++ 基本函数内的 200 多个实用基本函数。这些函数是作者近几年来在用 C 语言进行软件开发过程中不断积累、收集和精选出来的。它将教你如何实现汉字的输入及显示,如何控制键盘、鼠标和打印机,如何编制漂亮的图形交互式界面,以及如何实现数据的分析、管理等等。所有这些函数基本覆盖了实用软件开发过程中的大部分功能需要,有着极强的实用性和针对性。正确运用和恰当组合这些函数,你可创造出丰富多彩的新型软件。

为了方便你的学习,随书配有关于扩展函数库的软盘,其内容包括书中涉及的各种库文件、头文件和例子程序,以便你在自己的微机上迅速地掌握书中所述的各种功能,同时也可以直接利用这些函数搭制起你自己的应用软件。

全书包括如下五章内容:

第一章:“汉字的输入及显示”。介绍如何在屏幕上输入和显示汉字,以及设置汉字显示的字体、颜色、大小、方向等等。

第二章:“常用外部设备的控制”。介绍交互式信息输入过程中最常用的各种外部设备的控制函数,其中包括对键盘、鼠标、打印机的控制。

第三章:“Motif 风格的人机交互界面”。介绍人机交互式界面中两种最有用的工具——“按钮”和“窗口”。在此基础上,可以很方便地编制各种按钮式菜单、信息对话框、滚动框和信息输入表格等。这些功能对于各种交互式软件的开发是很有用的。

第四章:“Engine 数据库操作”。介绍输入计算机的信息如何存放、管理、统计等。

第五章:“统计图表”。主要介绍几个简单的统计图表函数,包括直方图、圆饼图和曲线图的绘制。

有关各章函数功能之间的基本关系如图 0.1 所示。

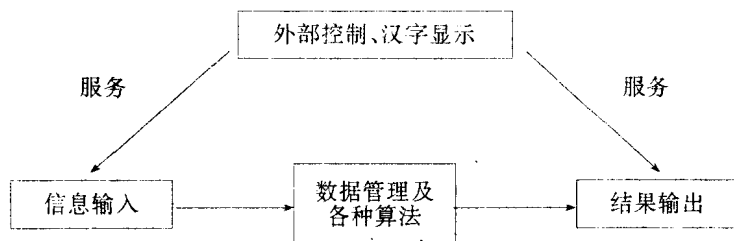


图 0.1 各章函数功能之间的基本关系

在书中,我们采用了面向对象的程序设计思想。对于每一类数,首先介绍这类函数所处理的对象的数据结构,然后逐一地分功能讲解函数。对于每一个函数,则按照功能、语法、使用说明、返回值和例子的顺序来介绍。

对于本书中的大部分函数,我们都提供了一个短小的例子来说明该函数的应用。这些例子都已在PC机上的Borland C++环境下执行通过。在与本书配套的软盘中提供了所有这些例子的源程序。

本书兼有示范性和实用性,既是程序学习员的参考资料,又是软件开发人员的辅助工具。

编 者

# 目 录

|                                  |    |
|----------------------------------|----|
| <b>第一章 汉字的输入及显示</b> .....        | 1  |
| 第一节 汉字的输入.....                   | 1  |
| 一、汉字输入中的两个全局结构.....              | 1  |
| 二、汉字输入的基本函数.....                 | 2  |
| 三、编制 getkey 函数时用到的底层函数.....      | 8  |
| 第二节 汉字的显示(一).....                | 11 |
| 一、初始化汉字系统.....                   | 12 |
| 二、汉字显示的基本函数.....                 | 12 |
| 三、设置汉字显示系统.....                  | 19 |
| 四、获取当前的汉字显示设置信息.....             | 31 |
| 五、关闭字库函数.....                    | 35 |
| 第三节 汉字的显示(二).....                | 36 |
| 一、汉字显示的基本函数.....                 | 36 |
| 二、设置、获取汉字系统环境.....               | 40 |
| 三、滚屏及其设置.....                    | 46 |
| <b>第二章 常用外部设备的控制</b> .....       | 49 |
| 第一节 鼠标器的使用.....                  | 49 |
| 第二节 打印机的使用.....                  | 54 |
| 一、初始化打印系统.....                   | 54 |
| 二、打印系统的设置.....                   | 56 |
| 三、打印控制基本函数.....                  | 62 |
| 四、屏幕硬拷贝.....                     | 65 |
| 第三节 键盘控制函数.....                  | 68 |
| <b>第三章 Motif 风格的人机交互界面</b> ..... | 72 |
| 第一节 按钮.....                      | 72 |
| 一、有关按钮的基本函数.....                 | 72 |
| 二、常用按钮函数.....                    | 74 |
| 第二节 按钮式菜单.....                   | 78 |
| 一、两个结构.....                      | 78 |

|                                   |            |
|-----------------------------------|------------|
| 二、三级菜单管理系统 .....                  | 81         |
| 三、菜单管理系统的基本函数 .....               | 85         |
| 第三节 对话框及基于对话框的函数 .....            | 92         |
| 一、对象的数据结构 .....                   | 92         |
| 二、对话框的创建和关闭 .....                 | 99         |
| 三、基于对话框的函数 .....                  | 101        |
| 四、数据场的输入 .....                    | 113        |
| 五、已有的专用对话框 .....                  | 120        |
| 第四节 滚动框 .....                     | 131        |
| 一、滚动框的数据结构 .....                  | 132        |
| 二、常用的滚动框函数 .....                  | 135        |
| 三、滚动框的编制技巧 .....                  | 144        |
| 四、编制函数 DisplayLine 时用到的底层函数 ..... | 147        |
| <b>第四章 Engine 数据库操作 .....</b>     | <b>150</b> |
| 第一节 Engine 的 C 接口 .....           | 150        |
| 第二节 数据库的初始化及终结 .....              | 151        |
| 第三节 数据表操作函数 .....                 | 157        |
| 一、打开、创建和关闭数据表函数 .....             | 157        |
| 二、对已有的未打开的数据表(或其家族)进行的基本操作 .....  | 162        |
| 三、测试一个已打开数据表信息的函数 .....           | 168        |
| 第四节 记录、字段操作函数 .....               | 174        |
| 一、当前记录的定位 .....                   | 175        |
| 二、操作记录缓冲区的基本函数 .....              | 180        |
| 三、记录缓冲区与数据表间数据交换的基本函数 .....       | 182        |
| 四、向记录缓冲区读写数据(字段级操作) .....         | 186        |
| 第五节 索引操作、搜索操作 .....               | 198        |
| 第六节 网络并发操作 .....                  | 205        |
| 一、文件及数据表锁 .....                   | 206        |
| 二、记录锁 .....                       | 209        |
| 三、数据表的并发使用 .....                  | 214        |
| 四、网络信息函数 .....                    | 215        |
| 第七节 口令操作 .....                    | 217        |
| 第八节 出错处理函数 .....                  | 220        |
| <b>第五章 统计图表 .....</b>             | <b>222</b> |
| 第一节 有关统计图表函数的一个结构 .....           | 222        |
| 第二节 统计图表函数 .....                  | 223        |

|                                   |     |
|-----------------------------------|-----|
| 附录.....                           | 232 |
| 附录 I 功能键及其扫描码、宏定义对照表 .....        | 232 |
| 附录 II Engine 错误代码表(按代码数值排列) ..... | 236 |
| 附录 III 软盘及其用法.....                | 240 |
| 参考文献.....                         | 244 |
| 函数索引.....                         | 245 |

## 第一章 汉字的输入及显示

本章介绍的是 C 语言扩展函数库中有关汉字的输入和显示部分的函数,它们的使用独立于任何汉字操作系统。运用这些函数工具,允许你在纯英文状态下方便地将汉字揉入到你的软件工作环境中,而不再需要借助 SuperDos、CCDos 一类的其它汉字操作系统。

本章将首先介绍汉字输入函数,然后介绍汉字显示函数。该扩展函数库中有两套完全独立的汉字显示函数:

- (1) 在各种显示卡下均可通用的汉字显示函数;
- (2) 专用于 VGA 显示卡的汉字显示函数。

汉字的输入和显示是编制各种中文软件的基础。

### 第一节 汉字的输入

本节介绍的汉字输入函数解决了通过键盘向计算机输入汉字的问题。为了帮助读者从汉字输入方法上能够更好地了解这些函数,我们首先介绍两个用于汉字输入的全局结构,以说明通过拼音和五笔字型输入汉字的主要思路。然后我们着重讲解在实际编程过程中常常需要调用的汉字输入函数。为了帮助读者能够自己编制汉字输入函数,在本书的最后将介绍在编制汉字输入函数过程中需要用到的一些基本函数。

#### 一、汉字输入中的两个全局结构

##### 1. 拼音输入的全局结构:

```
struct PYCODE {long value;  
               int code;  
               }PYcode [ ];
```

本全局结构数组主要用于存放拼音输入文件 py.dat,实现拼音输入显示功能。

长整型结构元素 value 的值跟键盘输入的拼音字母是相对应的。对于每一个输入的汉字拼音,例如:zhang,li,wang,通过一个算法,都可以找到一个唯一的 value 值与之对应。在本汉字输入系统中,算法大致如下:

```
value=0  
for(i=0;i<len;i++)  
    value+=(s[i]-‘a’+1)*Table[i];
```

其中整型变量 len 表示输入的汉字拼音的长度;字符型数组 s[ ]记录具体的每一个拼音字母是什么;整型数组 Table[ ]={1434890,531441,19683,729,2,0}。

整型结构元素 code 是跟 value 值相对应的区位码。根据它,即可到汉字库中读取相应的点阵信息,并将汉字显示在屏幕上。

## 2. 五笔字型输入的全局结构:

```
struct WBX {  
    char s[4];  
    int code;  
} *WBXcode;
```

本全局结构指针主要用于存放五笔输入文件 WBX.dat, 以实现五笔字型的输入显示功能。

字符型数组 s[4] 用于存放从键盘读入的五笔字型输入信息。

整型结构元素 code 是与 s[4] 相对应的区位码, 通过它即可找到所要求的汉字, 并使之显示在屏幕上。

## 二、汉字输入的基本函数

### load\_pycode

功 能: 将拼音输入文件 py.dat 调入内存。

语 法: void load\_pycode(void)

说 明: 在用拼音输入法输入汉字之前, 首先必须用本函数将拼音输入文件读入内存, 存放在全局结构 PYcode 中, 即一般所谓的调字库。

注 意: 函数 load\_pycode 默认 py.dat 文件在当前目录或其它可见的目录中。

返回值: 无

举 例: 参见函数 getkey 中的例子 em\_gtky.c

### load\_wbx

功 能: 将五笔字型输入文件 wbx.dat 调入内存。

语 法: void load\_wbx

说 明: 在用五笔字型输入法输入汉字之前, 首先必须用本函数将五笔字型输入文件读入内存, 存放在全局结构 WBXcode 中。

注 意: 与 load\_pycode 相同, 函数 load\_wbx 默认 WBX.dat 文件在当前目录或其它可见的目录中。

返回值: 无

举 例: 参见函数 getkey 例子 em\_gtkg.c

### getkey

功 能: 本函数根据输入功能键的不同, 可以按五笔字型、拼音、ASC II 码等各种方式从键盘读汉字或各种键值, 并显示在屏幕上。

语 法: int getkey(void)

说 明: 调用本函数前必须

- (1) 已经使用函数 load\_pycode, load\_wbx 将拼音或五笔字型输入文件调入内存;

(2) 屏幕处于图形方式下;

(3) 已用全局变量 vdisk 正确地指定了文件 cclib(16×16 点阵字库)和 system.fon(ASC II 码字库)的存放位置。

调用本函数后,将等待用户键盘输入。用户可以输入任一键值或 Shift,Alt,Ctrl 与任一键的组合键:

(1) Alt+F2——拼音输入

屏幕左下角将提示“拼音”,表示系统进入拼音输入工作状态,输入状态变量 KBstatus 被置为“KBS\_PY”。此时,系统将等待用户输入某汉字的拼音(1—7)个英文字母),显示出满足拼音的所有汉字,供用户选择,如图 1.1 所示:

拼 音:      dian      0:甸 1:店 2:惦 3:奠 4:淀 5:殿

图 1.1 汉字信息的输入

与一般汉字系统中的全拼音输入法完全一致,可以用如下的功能键帮助完成汉字的输入:

BACKSPACE : 取消因操作失误而输入的拼音字母;

ENTER : 取消当前已输入的拼音,以便重新开始输入;

BLANKSPACE : 根据输入的合法的汉字拼音,在屏幕底部显示出满足该拼音的全部汉字,以供用户选用;

“—”和“=” : 翻页功能键。“—”向前翻页,“=”向后翻页;

数字 0—9 : 当拼音输入后,可以敲入 0—9 之间的数字来选取所对应的汉字。

函数返回输入汉字的高位机内码、低位机内码被放在键盘缓冲区 KBbuffer 中返回。

## (2) Alt+F1——区位码输入

屏幕左下角将提示“区位码”，表示系统进入区位码输入工作状态，输入状态 KBstatus 被置为“KBS-QW”，此时系统将等待用户输入 4 个 0—9 之间的数字（前 2 个是区码，后 2 个是位码）。与一般的汉字系统中的区位码输入法相同，可以用如下的功能键来完成汉字的输入：

BACKSPACE : 取消因操作失误而输入的区位码数字；

ENTER : 取消当前已输入的区位码，以便重新开始输入；

BLANKSPACE : 按下该键后将根据输入的合法的区位码在屏幕上显示出相应的汉字或 ASC I 码或其它图形符号，以完成一个区位码的输入操作。

函数返回输入的汉字或其它图形符号的高位机内码，低位机内码被放入键盘缓冲区 KBbuffer 中返回。

## (3) Alt+F4——五笔字型输入

屏幕左下角将提示“五笔字型”，表示系统进入五笔字型输入工作状态，输入状态变量 KBstatus 被置为“KBS-WB”。此时，系统将等待用户输入某汉字的五笔字型代码（1—4 个英文字母）。与一般的汉字系统中的五笔字型输入法完全相同，可以用如下的功能键来完成汉字的输入：

BACKSPACE : 取消因操作失误而输入的字母。

ENTER : 取消当前已输入的五笔字型代码，以便重新开始输入；

BLANKSPACE : 根据输入的合法的五笔字型代码，在屏幕底部显示出相应的全部汉字。

函数返回输入汉字的高位机内码，汉字的低位机内码被放入键盘缓冲区 KBbuffer 中返回。

## (4) Alt+F10——西文输入

屏幕左下角将提示“西文”，表示系统进入西文输入工作状态，输入状态变量 KBstatus 被置为“KBS-ASC I”。此时，系统将等待用户敲入键盘上的任一键，或任一键与 Shift, Alt, Ctrl 的组合键，函数返回该键或组合键的扫描码。附录 I 给出了这些键的宏定义及相应的扫描码值。

## (5) Ctrl+s——设置打印机转换色彩

按下该键后屏幕上将出现一个提示对话框。用户可用 0—9 以及 A, B, C, D, E, F 这 16 个键设置打印机的色彩转换，供硬拷贝屏幕时使用。具体如何设置打印机色彩，参见函数 GetPrnParams。

此时函数返回 Ctrl+s 的扫描码。

## (6) Ctrl+p——屏幕拷贝

按下本键后，将对屏幕进行硬拷贝。正确地使用该命令必须确保：

(a) 系统中已联接了打印机；

(b) 已正确地选定了打印机的类型。

此时，函数返回 Ctrl+p 的扫描码。

(7) 按下除上述各键以外的任意键

函数将按默认的“西文”输入方式返回输入键或组合键的扫描码。功能基本同西文输入状态,只是:

- (a) 屏幕底部没有“西文”输入的提示;
- (b) Alt 与 a、b...z 的组合键失效。

在调用本函数时还将影响到如下几个全局性变量:

(1) int KBstatus

根据当前的键盘输入状态,可取值:

|   |           |      |
|---|-----------|------|
| 0 | KBS_ASC I | 西文   |
| 1 | KBS_QW    | 区位码  |
| 2 | KBS_PY    | 拼音   |
| 3 | KBS_WB    | 五笔字型 |
| 4 | KBS_CC    | 层次四角 |

(2) int KBbuffer [16] 键盘输入缓冲区

int KBcount 指向当前键盘缓冲区位置的指针

返回值:略(见“说明”部分)。

举 例:em\_gtky.c

```
/* -----
                                em_gtky.c
usage example of function int getkey(void)
----- */

#include <graphics.h>
#include <stdio.h>

#define KBS_ASC I      0
#define KBS_QW         1
#define KBS_PY         2
#define KBS_WB         3
#define KBS_CC         4

char vdisk = 'c';
extern int KBcount,KBbuffer[16];
extern int KBstatus;

void main( )
{
    int x=0,y=0;
    int key,kkey;
    int gdrive = DETECT, gmode;
    unsigned char c1,c2;
    char ss[5];
```

```

load_pycode( );
load_wbx( );

initgraph(&gdrive,&gmode,"c:\dot");
do {
    key = getkey();
    c1 = (unsigned char) key;
    if( KBstatus == KBS_ASCⅡ ) {
        sprintf(ss,"%c",c1);
        mPutMessage(ss,x,y);
        x++;
    }
    else if( KBcount > 0 ) {
        kkey = KBbuffer[KBcount-1];
        KBcount--;
        c2 = (unsigned char) kkey;
        sprintf(ss,"%c%c",c1,c2);
        mPutMessage(ss,x,y);
        x = x+2;
    }
    if( x >= 80 ) { x=1;y++;}
} while ( key != 27 );
closegraph();
}

```

### putkey1

**功 能：**将调用函数 getkey 所获得的汉字、图形符号或 ASCⅡ 码显示到屏幕上。

**语 法：**void putkey1(int key, int x, int y)

**说 明：**本函数最好紧跟在函数 getkey 之后,因为需要用到 getkey 中产生的以下 3 个信息:

1. KBstatus: 本函数只有当 KBstatus==KBS\_ASCⅡ 时,才能显示 ASCⅡ 码,否则将只能显示汉字信息;
2. getkey 返回的 ASCⅡ 扫描码或汉字的高位码;
3. 从键盘缓冲区 KBbuffer 返回的汉字低位码。

其中 1、3 两项信息在函数内部自动传递,第二项信息用 getkey 中的返回值,通过函数的输入参数 key 送入 putkey1 中。参数(int x, int y)确定字符或 ASCⅡ 码在屏幕上的显示位置,以字节为单位,每个字节的长度为 8 个像素单位。x 确定水平方向的位置,y 确定垂直方向的位置。对于分辨率为 640×480 的 VGA 图形卡,x,y 的取值范围应该为:

$$x \in [0,79] \quad y \in [0,59]$$

返回值: 无

举 例: putkey.c

```
/*    putkey.c    */
#include <graphics.h>
#include <stdio.h>
#include <keys.h>

#define KBS_ASCII 0
#define KBS_QW 1
#define KBS_PY 2
#define KBS_WB 3
#define KBS_CC 4

char vdisk = 'c';
extern int KBcount, KBbuffer[16];
extern int KBstatus;

void main( )
{
    int x=0,y=0;
    int key;
    int gdrive = DETECT, gmode;

    load_pycode( );
    load_wbx( );

    initgraph(&gdrive,&gmode," ");

    do {
        key = getkey( );
        putkey1(key,x,y);
        if ( KBstatus == KBS_ASCII) x++;
        else if (KBcount > 0) x = x+2;
        if (x >= 80) { x=1; y++; }
    } while (key != 27);
    closegraph( );
}
```

### putkey2

功 能: 将调用函数 getkey 所获得的汉字、图形符号或 ASC II 码显示到屏幕上。

语 法: void put key2(int key, int x, int y)

说 明: 本函数的用法与 putkey1 基本相同,只是变量 x,y 以像素为单位。对于分辨率为 640×480 的 VGA 图形卡,x,y 的取值范围应该为:

$x \in [0, 639]$      $y \in [0, 479]$

返回值：无

举 例：(略)

### **free\_wbx**

功 能：释放调用五笔字型输入文件所占用的内存。

语 法：void free\_wbx(void)

说 明：本函数释放全局结构指针 WBXcode 所占用的内存。由于拼音输入文件所占用的内存较小，故本函数库中未定义专门的内存释放函数。

返回值：无

举 例：(略)

## **三、编制 getkey 函数时用到的底层函数**

在前面我们看到函数 getkey 包含了多种功能，它主要由如下一些基本函数组成：

### **kPrompt**

功 能：在屏幕底部显示输入提示符。

语 法：void kPrompt(void)

说 明：本函数在屏幕底部显示要求输入信息的提示，如下图所示：

××：

其中××可以为：西文、区位码、拼音或五笔字型。具体内容全局变量 KBstatus 决定。

返回值：无。

举 例：em\_kprom.c

```
/* -----  
                                     em_kprom.c  
usage example of function void kPrompt (void)  
----- */  
#include <graphics.h>  
  
char vdisk = 'c';  
extern int KBstatus;  
  
void main()  
{  
    int gdrive = DETECT, gmode;  
  
    initgraph(&gdrive, &gmode, "c:\\dot");  
    for( KBstatus = 0; KBstatus < 5; KBstatus++) {  
        . 8 .  
    }
```

```

    kPrompt();
    getch();
}
closegraph();
}

```

## KChoices

**功 能：**在拼音输入过程中，根据输入的拼音在屏幕底部显示出相对应的全部汉字，以供用户选择。

**语 法：**int KChoices (int num, int lnext)

**说 明：**本函数需要输入如下信息：

1. 整型参数 num, 用于确定与输入拼音相对应的汉字在结构数组 PYcode 中的位置, 它与拼音是一一对应的。num 的取值范围与调入内存的汉字输入库的大小有关。对于一级汉字库, num 应在 0—400 之间。若 num = -1, 则表示在屏幕底部显示一个空行。

2. 整型变量 lnext, 由于对应于同一个拼音可以有多个汉字, 当这组汉字多于 10 个时, 需要分成多页显示。lnext 就是用于确定所显示出的该页上第一个汉字在该组汉字中的相对位置。

由上可知, 屏幕底部显示出的第一个汉字在结构数组 PYcode 中的位置为 num, 并且通过 lnext 对应结构 PYcode[num] 中的变量 code, 从而得到该汉字的区位码。

**返回值：**若 num = -1, 则返回值为 -1;

否则, 返回所显示出的最后一个汉字的机内码。

(注: 机内码 code 与区码 C1、位码 C2 的对照关系如下:

$$C1 = code / 100 + 0xa0 - 1$$

$$C2 = code \% 100 + 0xa0$$

**举 例：**em\_kchoi.c

```

/* -----
                                     em_kchoi.c
usage of function: int KChoices(int num, int lnext)
    num: determinate the location of first hanzi in PY.DAT
    lnext: determinate how many hanzi shown next
    if num is not existed in PY.DAT return -1
    else return the hanzi code of the last hanzi
----- */
#include <graphics.h>

char vdisk = 'c';

void main()

```