

# 第 1 部分 基础实验

“C 语言程序设计”是一门实践性特别强的课程，必须重视实践环节。目前，计算机水平考试多为上机考试形式，死记硬背一些语法要点和基本概念已经无法达到本课程的教学要求。因此，为了学好本门课程必须加强上机实践，即独立编写程序，独立上机调试程序。

不少人认为上机实验就是验证教材上的举例，看运行结果是否与书上的结果一致。其实，上机实验不仅包括验证性实验、设计性实验，还包括综合性实验和创新实验，因此，上机实验具有如下更广泛的目的：

(1) 加深理解。C 语言的语法较多、使用灵活且概念较抽象，仅仅依靠课堂讲授是远远不够的。上机操作可以加深对讲授内容的理解，尤其是一些语法规定和容易出错的地方。

(2) 提高操作水平。上机实验必须了解和熟悉 C 程序设计的开发环境，譬如 Turbo C 软件的安装、设置和使用。由于计算机软件的开发环境功能和操作方法具有一定的共性，因此，熟悉、掌握了一种开发环境对学习新的开发环境具有触类旁通的作用。

(3) 锻炼独立解决问题的能力。C 语言的初学者肯定会碰到各种各样的问题，有些读者习惯碰到问题就请教别人，其实，这种习惯不一定好，因为它不能锻炼自己独立解决问题的能力。为了提高自己解决问题的能力，碰到问题时首先要认真分析错误提示信息，争取自己排除错误。这种方法在开始的时候上机效率可能不高，但独立解决问题的能力锻炼出来后，完全可以达到事半功倍的效果。

(4) 养成良好的学习习惯。每一次上机实验都应该做好准备，提前预习，上机过程中要做好记录，上机结束后及时复习和总结。养成良好的学习习惯不仅可以提高本课程的学习效果，对人们今后的学习和工作也是十分有益的。

第 1 部分是为了使读者熟练掌握 C 语言的语法和结构化程序设计思想而设计的基础实验，这些实验包括验证性、设计性实验共 16 个，每个实验包括如下 5 个部分：

- (1) 实验目的与要求；
- (2) 实验预习与准备；
- (3) 实验内容与步骤；
- (4) 实验评价；
- (5) 思考与讨论。

## 实验 1 熟悉 Turbo C 的上机环境

### 【实验目的与要求】

- 了解 Turbo C 的安装、启动和退出；
- 了解 Turbo C 菜单的用法；
- 熟练掌握编辑、编译和运行 C 程序的一般步骤；
- 掌握 Turbo C 常用快捷键的用法。

### 【实验预习与准备】

#### 1. Turbo C 的安装

目前获得的 Turbo C 软件安装时，只需将 Turbo C 目录拖到指定的磁盘分区，譬如 D:。

#### 2. Turbo C 的启动

Turbo C 的可执行程序为 `tc.exe`，因此，只要找到该文件就可以通过双击该文件图标来启动；为了操作方便，可以首先建立该文件的快捷方式，再将该快捷方式拖到桌面上，以后每次启动可以直接双击快捷方式图标。

#### 3. Turbo C 的关闭

要从 Turbo C 环境退出，可以用鼠标点击窗口关闭按钮，也可以按快捷键 `Alt + X`，还可以通过菜单方式选择退出。

#### 4. C 程序上机的一般步骤

C 程序上机的一般步骤包括编辑、编译、连接和运行等过程，上机调试程序的流程如图 1-11 所示。

#### 5. Turbo C 的上机操作

Turbo C 的上机操作包括两种方式，一种方式采用菜单方式，读者对这种方式应该比较了解；另一种方式是快捷键方式，为了更好地完成上机操作，建议读者熟练掌握表 1-11 所示的快捷键。

表 1-11 快捷键及其功能

快捷键名	功 能
F1	获取帮助
Ctrl + F1	获取特定内容的帮助
F2	保存
F9	编译
Ctrl + F9	编译并连接
Alt + F5	查看用户屏幕，即屏幕输出结果
F3	加载源程序，即将源程序调入内存
Alt + F3	从最近使用过的文件列表中挑选源程序

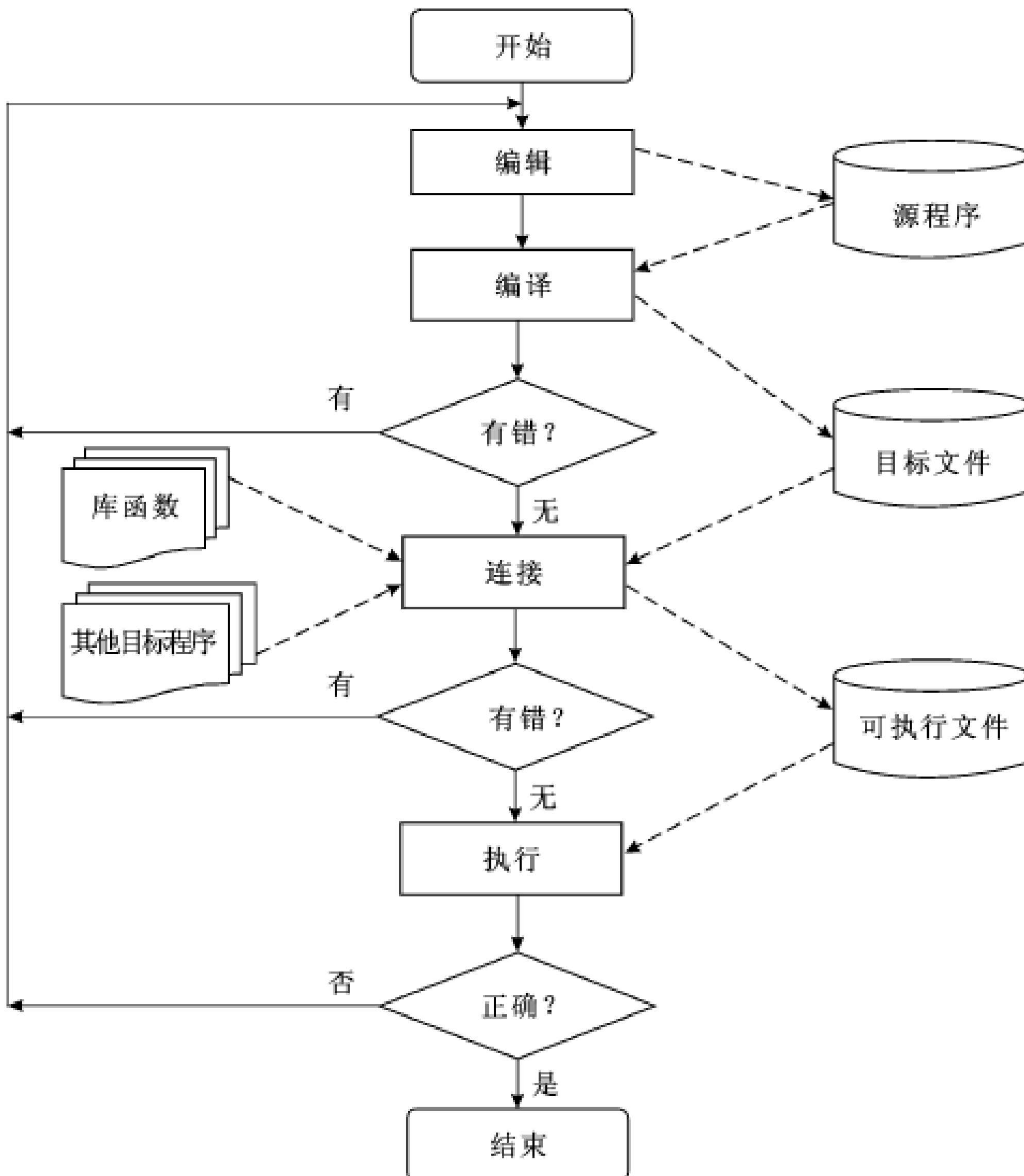


图 1 11 上机调试程序的流程

## 【实验内容与步骤】

1. 在计算机硬盘的 D 盘以你的姓名缩写建立一个文件夹（注：长度不超过 8 个英文字符）。

查看本计算机的 Turbo C（以下简称 TC）所安装的目录，记录下安装的目录。

TC 所在目录是：\_\_\_\_\_。

2. 启动 TC，进入 TC 界面。

按 F10 键进入主菜单，如图 1 12 所示，将光标移动到 Options 菜单，选择下面的 Directories 菜单项，检查其参数特性，并保证 Include directories、Library directories 和 Turbo C directory 的路径都是前述第 1 部分所查看到的目录，否则，需要修改参数配置。

例如，若 TC 安装在 D:\TC20 下，则 Include directories 的路径应当是 D:\TC20\INCLUDE，Library directories 的路径应当是 D:\TC20\LIB，Turbo C directory 的路径应当是 D:\TC20。

注意：如果系统配置参数发生变化，最好用 Options 菜单下的 Save options 菜单项进行保存，否则退出 TC 时参数没有保存，待下次启动时，还要重新设置这些参数。设置参数的界面如图 1 12 所示。



图 1 12 设置参数的界面

3. 上机输入程序，并回答问题。

```
#define PI 3.14
main()
{
    float r, s;
    printf( " \nr = " );
    scanf( " %f", &r);
    s = PI 3 r 3 r;
    printf( " Circle area = % . 2f", s);
}
```

回答下列问题：

- |                                |                                |                                  |                                |
|--------------------------------|--------------------------------|----------------------------------|--------------------------------|
| (1) 文件编辑后，用 ex0101 1c 保存，是否成功？ | <input type="checkbox"/> 成功    | <input type="checkbox"/> 不成功     |                                |
| (2) 进行编译，有错误吗？                 | <input type="checkbox"/> 有     | <input type="checkbox"/> 没有      |                                |
| (3) 如果有错误，有几个错误？               | <input type="checkbox"/> 2 个以下 | <input type="checkbox"/> 2 ~ 5 个 | <input type="checkbox"/> 5 个以上 |
| (4) 能确认是哪些行有错吗？                | <input type="checkbox"/> 能     | <input type="checkbox"/> 不能      |                                |
| (5) 程序修改正确了吗？                  | <input type="checkbox"/> 正确    | <input type="checkbox"/> 不正确     |                                |
| (6) 能观察程序运行的结果吗？               | <input type="checkbox"/> 能     | <input type="checkbox"/> 不能      |                                |
| (7) 关闭此文件后，能再次调入 ex0101 1c 吗？  | <input type="checkbox"/> 能     | <input type="checkbox"/> 不能      |                                |

4. 输入下面的程序，以 ex0102 1c 文件名保存，分析程序功能。

```
main()
{
    int a, b, c;
    printf( " \n input a, b: " );
    scanf( " %d, %d", &a, &b);
```

```
    c = max( a, b ) ;
    printf( " \n max( % d, % d) value = % d" , a, b, c) ;
}
int max( int x, int y)
{
    int z;
    if( x < y) z = y;
    else z = x;
    return z;
}
```

运行结果:

输入: \_\_\_\_\_。

输出: \_\_\_\_\_。

程序功能: \_\_\_\_\_。

5. 指出下面程序中的错误, 并在上机过程中予以更正。

```
1:  main( )
2:  {
3:      int a;
4:      scanf( " % f" , a) ;
5:      printf( " a = % f" , a) ;
6:      char c;
7:      c = A;
8:      printf( " % f" , c)
9:  }
```

6. 输入并运行下面一个需要在运行时输入数据的程序, 要求从键盘输入 3 个实数, 输出它们的和。

```
1:  #include < stdio. h >
2:  main( )
3:  {   float a, b, c, sum;
4:      printf( " input a, b, c: " )
5:      scanf( " % f % f % f" , &a, &b, &c) ;
6:      sum = a + b + c;
7:      printf( " \n sum = % f\n" , sum) ;
8:  }
```

## 【实验评价】

实验任务评价表

题号	完成情况	知识点掌握情况	难度系数	测试评价
1		启动和关闭 TC 环境	1	<input type="checkbox"/> 了解 <input type="checkbox"/> 掌握 <input type="checkbox"/> 应用
2		TC 中文件的打开、保存	1	<input type="checkbox"/> 了解 <input type="checkbox"/> 掌握 <input type="checkbox"/> 应用
3		配置 TC 的基本参数	2	<input type="checkbox"/> 了解 <input type="checkbox"/> 掌握 <input type="checkbox"/> 应用
4		源文件的编辑、编译和运行	2	<input type="checkbox"/> 了解 <input type="checkbox"/> 掌握 <input type="checkbox"/> 应用
5		程序的查错与定位	3	<input type="checkbox"/> 了解 <input type="checkbox"/> 掌握 <input type="checkbox"/> 应用

注：难度系数为 1 ~ 5，其中：1 为容易，5 为难。

## 【思考与讨论】

1. 总结一下完成一个完整的 C 程序设计的过程。
2. 能否在一个文件中编制两个源程序(即有两个 main 函数)? 如果不能应该如何做?
3. 将本实验过程中碰到的问题和出错信息记录下来, 并与其他同学讨论。

## 实验 2 基本数据类型

### 【实验目的与要求】

- 了解 Turbo C 的基本编辑快捷键；
- 了解窗口的切换功能键；
- 掌握获取联机帮助的方法；
- 熟练掌握整数的表示形式、单个字符与字符串常量的表示形式；
- 理解并掌握格式控制符 %d、%c、%f 的基本用法。

### 【实验预习与准备】

#### 1. 快捷键

为了提高编辑程序的效率，Turbo C 2.0 集成环境提供了许多编辑快捷键，常用的编辑快捷键如表 2.11 所示。

表 2.11 Turbo C 2.0 常用的编辑快捷键

PageUp	向前翻页	Ctrl + KB	设置块开始
PageDn	向后翻页	Ctrl + KK	设置块结尾
Home	将光标移到所在行的开始	Ctrl + KV	块移动
End	将光标移到所在行的结尾	Ctrl + KC	块拷贝
Ctrl + Y	删除光标所在行	Ctrl + KY	块删除
Ctrl + T	删除光标所在处的一个词	Ctrl + KP	块文件打印
Ctrl + KR	读文件	Ctrl + Q [	查找 Turbo C 2.0 双界符的后匹配符
Ctrl + KW	存文件	Ctrl + Q ]	查找 Turbo C 2.0 双界符的前匹配符
Ctrl + F1	如果光标所在处为 Turbo C 2.0 库函数，则获得有关该函数的帮助信息	F1	获得帮助信息

#### 2. 窗口切换功能键

Turbo C 2.0 的主窗口包括主菜单行、编辑窗口和消息窗口。主菜单提供所有的 TC 操作命令，编辑窗口提供源程序输入界面，消息窗口则显示编译或调试时可能出现的错误信息。这三个窗口之间的切换过程如图 2.11 所示。

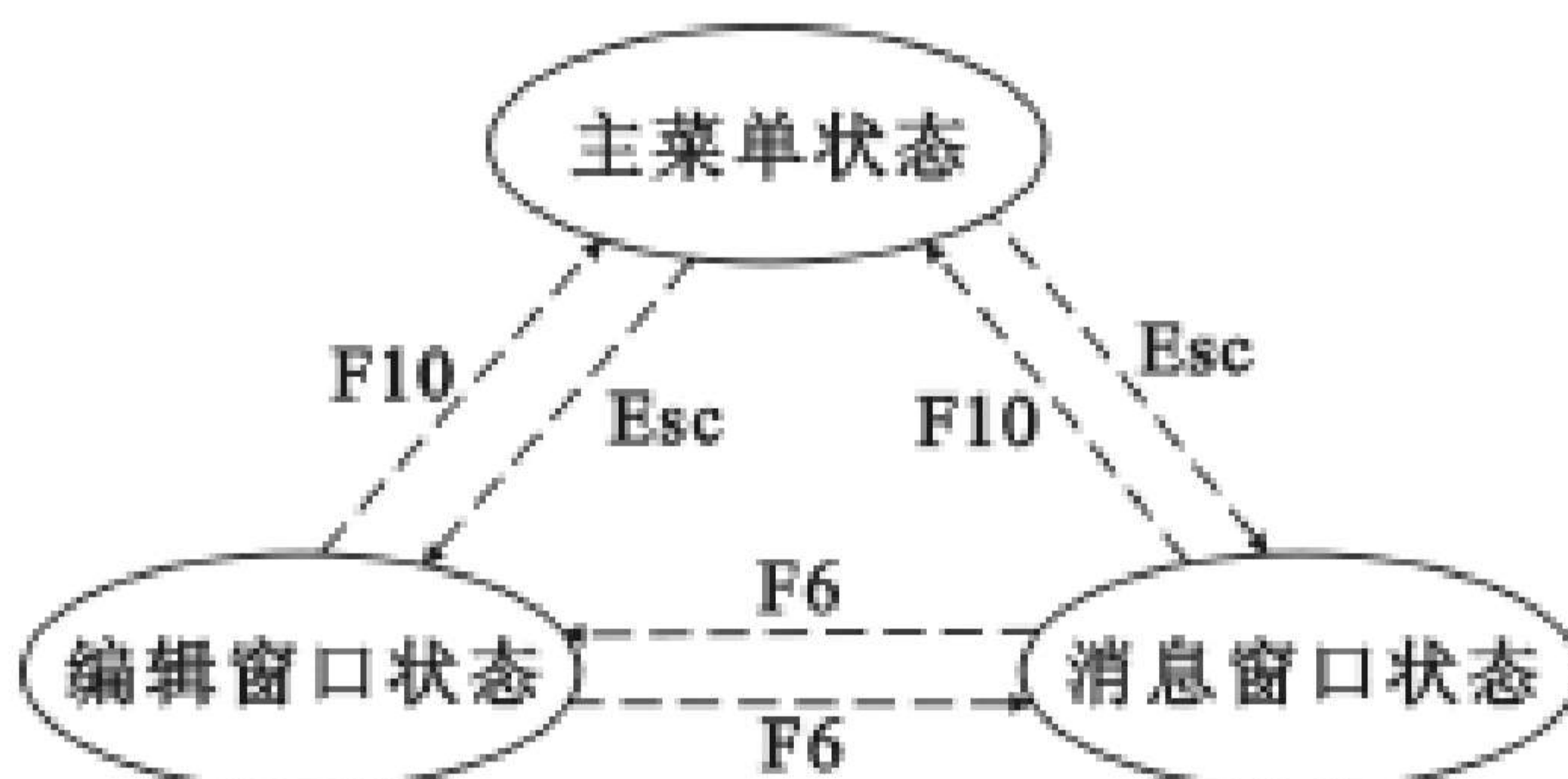


图 2.11 窗口之间的切换过程

### 3. 获取帮助

Turbo C 2.0 具有比较完善的联机帮助功能，充分利用这些功能是熟悉和掌握 C 程序设计的捷径。TC 提供了两种联机帮助快捷键。

(1) F1 键：获取帮助信息，一般的软件系统均默认 F1 键为获取帮助的功能键。在编辑状态下按 F1 键，可以查看关于编辑器的全部编辑命令的使用方法帮助信息；在某个菜单状态下按 F1 键，会出现关于此菜单项的功能作用的解释。编辑命令的帮助信息界面如图 2.12 所示。

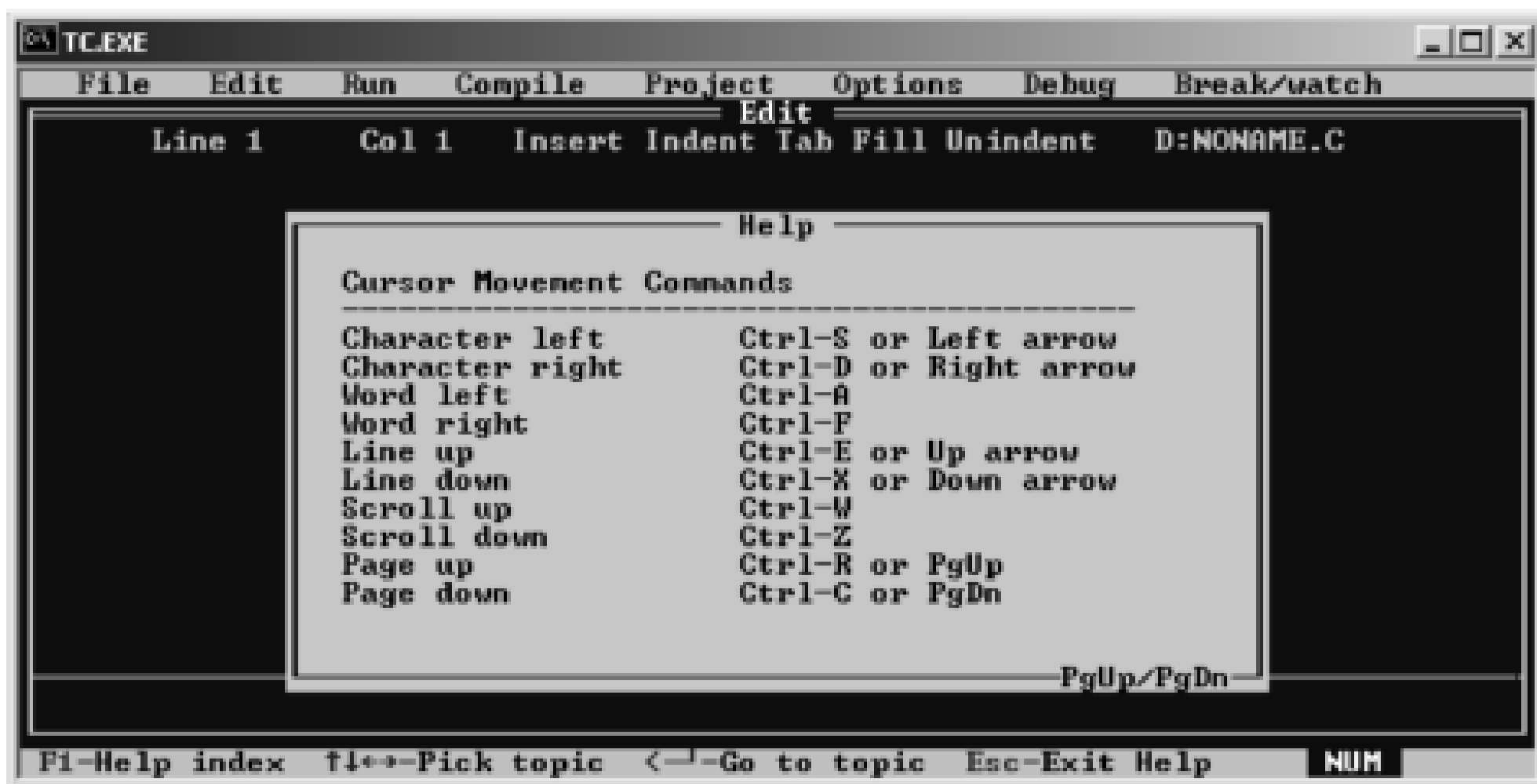


图 2.12 编辑命令的帮助信息界面

(2) Ctrl + F1 键：可以获取关于 C 语言库函数的有关帮助信息。如果光标所在处为 Turbo C 2.0 库函数，则获得有关该函数的帮助信息，如图 2.13 所示。

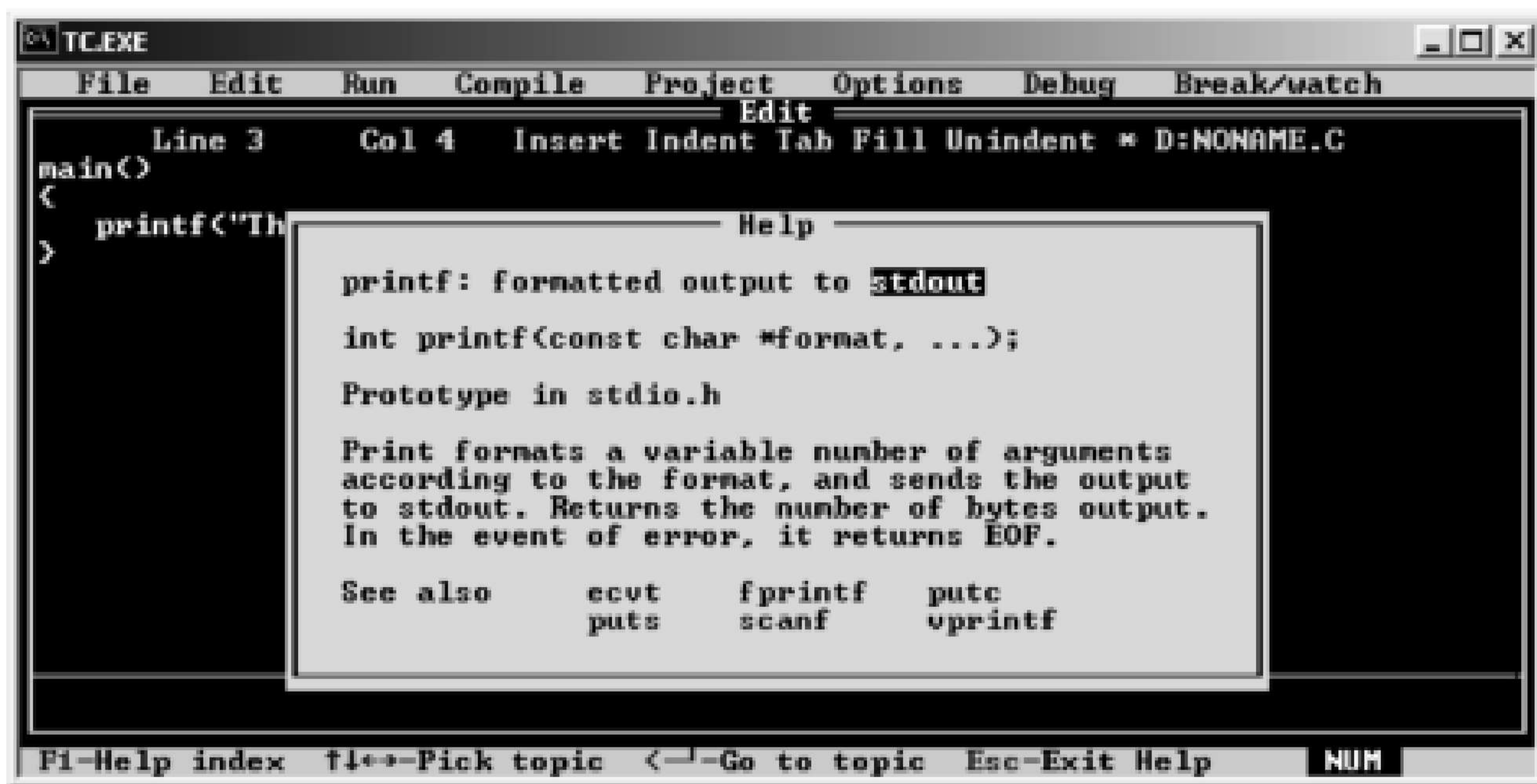


图 2.13 库函数的帮助信息界面

### 4. 填空题

(1) 16 位 PC 机中，一个 int 变量占\_\_\_\_个字节，一个 char 变量占\_\_\_\_个字节，一个 unsigned int 变量占\_\_\_\_个字节，一个 float 变量占\_\_\_\_个字节，一个 double 变量占\_\_\_\_个字节，一个 long int 变量占\_\_\_\_个字节。

- (2) 十进制数 100, 转换成八进制表示为\_\_\_\_, 用十六进制表示为\_\_\_\_。
- (3) C 语言中, 字符常量是用\_\_\_\_引起来, 字符串常量是用\_\_\_\_引起来。
- (4) '\t'表示\_\_\_\_, '\\ '表示\_\_\_\_, '\b'表示\_\_\_\_, '\n'表示\_\_\_\_。
- (5) 一个整数与一个字符进行运算, 则自动转换成\_\_\_\_进行运算; 一个整数与浮点进行运算, 则自动转换成\_\_\_\_进行运算; 两个字符进行运算, 则结果的类型是\_\_\_\_; 长整型与浮点型进行运算, 则转换成\_\_\_\_进行运算。

### 【实验内容与步骤】

1. 启动 TC 环境, 编辑程序以 ex0201 lc 文件名保存。

```

1:  main( )
2:  {
3:      int i;
4:      i = - 2;
5:      printf( " dec = % d, oct = % o, hex = % x, unsigned = % u\n" , i, i, i, i);
6:  }
```

- (1) 第一次编译有错误吗?      有      没有
- (2) % d、% o、% x、% u 分别表示何种控制格式?
- (3) 运行结果: \_\_\_\_\_。

2. 分析程序运行结果。

```

1:  void main( )
2:  {
3:      printf( " How are you? \n" );
4:      printf( " I am fine. \n\n" );
5:
6:      printf( " How are you? \t" );
7:      printf( " I am fine. \n\n" );
8:
9:      printf( " How are you? \n" );
10:     printf( " \bI am fine. \n\n" );
11:
12:     printf( "I am fine. " );
13:     printf( " \rHow are you? \n\n" );
14: }
```

- (1) 解释 '\n' 的实际控制的输出结果。
- (2) 解释 '\t' 的实际控制的输出结果。
- (3) 解释 '\b' 的实际控制的输出结果。
- (4) 解释 '\r' 的实际控制的输出结果。
3. 编程测试下面程序的输出结果, 并回答问题。

```

1:  main( )
```

```

2:  {
3:     int a, b;
4:     float c, d;
5:     char c1, c2;
6:     a = 61; b = 62;
7:     c = 31.5; d = -68.87;
8:     c1 = 'a'; c2 = 65;
9:     printf( " \na = %d, a - b = %d\n" , a, a - b );
10:    printf( " c = %4.2f, d = %5.2f\n" , c, d );
11:    printf( " c = %e, d = %e\n" , c, d );
12:    printf( " c1 = %c, %c\n" , c1, c2 + 1 );
13: }

```

回答下列问题:

- (1) 第 3 行, int a, b; 变量 a、b 之间能否用分号隔开? 为什么?
- (2) 第 6 行, 此行有几个语句? 如何判断?
- (3) 变量 a 与变量 c 能否进行 a - c 的运算? 说明理由。
- (4) 第 10 行, %4.2f 表示什么意思?
- (5) 第 11 行, %e 表示什么意思?
- (6) 运行结果: \_\_\_\_\_。

4. 程序改错题。找出程序中的错误并改正, 回答题后的问题。

从键盘输入一个华氏温度值, 要求输出其对应的摄氏温度值, 精确到小数点后两位。

转换公式如下:

$$c = \frac{5}{9}(f - 32)$$

```

1:  main( )
2:  { float f, c;
3:     scanf( " %d" , &f );
4:     c = 5/9 * ( f - 32 );
5:     printf( " f = %5.2f, c = %5.2f" , &f, &c );
6:  }

```

回答下列问题:

- (1) 上述程序有多处错误, 请指出并改正。
- (2) 测试运行结果:  
 输入 0 时, 输出为\_\_\_\_\_;  
 输入 32 时, 输出为\_\_\_\_\_;  
 输入 100 时, 输出为\_\_\_\_\_;  
 输入 -50 时, 输出为\_\_\_\_\_。
- (3) 以上测试结果对程序有影响吗? 如果有, 说明理由。
- (4) 上述程序健壮吗? 如果不健壮, 表现在哪里? 通过什么手段可以使之变得健壮?

5. 输入并运行下面程序，对照程序分析运行结果，并回答问题。

```

1:  #include <stdio. h >
2:  main( )
3:  {   int a, b;
4:      long d, e;
5:      unsigned f, g;
6:      char c;
7:      a = 100;   b = - 100;
8:      c = 'a';   d = 50000;
9:      e = 32767;
10:     printf( " % d, % d \n" , a, b) ;
11:     printf( " % ld, % ld \n" , d, e) ;
12:     printf( " % c, % d \n" , c, c) ;
13:     f = a;   g = b;
14:     printf( " % u, % u \n" , f, g) ;
15:     a = c;
16:     printf( " % d, % d \n" , a, c) ;
17:     a = d;   b = e;
18:     printf( " % d, % d \n" , a, b) ;
19:  }
```

回答下列问题：

- (1) 将一个大于 32767 的长整型数赋给整型变量，会得到什么结果？
- (2) 将一个负整数赋给一个无符号的变量，会得到什么结果？
- (3) 将一个字符型变量赋给一个整型变量，会得到什么结果？

6. 编程题

从键盘输入一个半径值  $r$ ，求圆的面积并输出结果。

### 【实验评价】

实验任务评价表

题号	完成情况	知识点掌握情况	难度系数	测试评价
1		整数的不同码制	2	<input type="checkbox"/> 了解 <input type="checkbox"/> 掌握 <input type="checkbox"/> 应用
2		输出格式控制	1	<input type="checkbox"/> 了解 <input type="checkbox"/> 掌握 <input type="checkbox"/> 应用
3		基本数据输出控制格式	2	<input type="checkbox"/> 了解 <input type="checkbox"/> 掌握 <input type="checkbox"/> 应用
4		输入/输出、表达式计算	2	<input type="checkbox"/> 了解 <input type="checkbox"/> 掌握 <input type="checkbox"/> 应用
5		数据类型转换	4	<input type="checkbox"/> 了解 <input type="checkbox"/> 掌握 <input type="checkbox"/> 应用
6		综合应用	4	<input type="checkbox"/> 了解 <input type="checkbox"/> 掌握 <input type="checkbox"/> 应用

注：难度系数为 1 ~ 5，其中：1 为容易，5 为难。

## 【思考与讨论】

## 1. 选择题

- (1) 语句 `int a, b; scanf(“%d%d”, &a, &b);` 中不能作为输入分隔符的是( )。
- A. ,            B. 空格            C. 回车键            D. Tab 键
- (2) C 语言中, `int` 型数据在内存中的存储形式是( )。
- A. ASCII 码    B. 原码            C. 反码            D. 补码
- (3) 执行 `printf(“%x”, 100);` 语句后的输出结果是( )。
- A. 100            B. 0x100            C. 0x64            D. 64

## 2. 分析程序运行结果

```
main( )
{   char a = 1;
    int p;
    p = 123;
    p = 0 - a;
    printf(“p = %u”, p);
}
```

运行结果: \_\_\_\_\_。

3. 一种数据譬如 `int x = 33;`, 能否按不同格式输出?
4. 在程序中, 语句 `printf(“%c”, “a”);`, 会出现哪种错误信息? 请与同学讨论。
5. 如果要计算  $10!$ , 则保存结果应该定义为什么类型?

## 实验 3 运算符与表达式

### 【实验目的与要求】

掌握标识符的定义方法；  
理解表达式的含义；  
理解运算符的优先级和结合方向；  
理解并掌握表达式的计算方法。

### 【实验预习与准备】

1. 标识符的组成: \_\_\_\_\_。

2. 表达式的构成: \_\_\_\_\_。

3. 算术运算符的优先级和结合方向: \_\_\_\_\_。

4. 自增、自减运算符与赋值运算的关系: \_\_\_\_\_。

5. 从下列 10 个有关运算符和表达式的描述中, 选择 5 个正确答案: \_\_\_\_\_。

(1) C 语言中, 一个 `int` 型数据在内存中占 2 个字节, 而 `unsigned int` 型数据在内存中占 4 个字节。

(2) C 语言中的变量必须遵循“先定义, 后使用”的原则。

(3) 取余运算 `%` 要求参加运算的两个数必须都是整数。

(4) `=` 运算和 `==` 运算是具有相同级别的运算, 按照从右到左的规则进行运算。

(5) 若 `a` 和 `b` 的类型相同, 在执行了赋值表达式 `a = a + b` 后, `b` 中的值不变。

(6) 执行程序段 `int a = 1, b = 2, c = 3; c = a > b ? a, b;` 后, `c` 的值是 3。

(7) 一个 `int` 变量的取值范围是  $-32767 \sim 32768$ , 一个 `unsigned int` 变量的取值范围是  $0 \sim 65535$ 。

(8) `sizeof (float)` 是一个整型表达式, 而不是一个函数调用。

(9) 执行程序段 `int a, b = 5; a = (b = 3);` 后, `a` 的值是 1。

(10) C 语言默认浮点数精确到小数点后 6 位, 则  $1 \ 10 + 12 / 5 \ 3 \ 2$  的值是 5 1000 000。

### 【实验内容与步骤】

1. 输出控制格式, 分析程序运算结果, 并回答问题。

```
1: main( )
2: {   char c1, c2;
3:     c1 = 'a'; c2 = 'b';
4:     printf( " %d, %c\n", c1, c1);
5:     printf( " %d, %c\n", c2, c2);
```

```
6: }
```

回答下列问题:

(1) 第 4 行, 控制格式 %d 和 %c 分别表示什么含义?

(2) 第 4 行, 为什么对于同样的 c1 进行输出时, 格式不同会产生不同的输出结果? 请说明理由。

(3) 运行结果: \_\_\_\_\_。

(4) 第 2 行, 将 char c1, c2; 变成 int c1, c2;, 程序对吗? 为什么?

2. 算术运算。输入一个小于 1000 的整数, 编程求出个位数、十位数和百位数并输出, 并回答问题。

```
1: main( )
2: {
3:     int x, a, b, c;
4:     printf( " \ninput x ( x < 1000) : " );
5:     scanf( " %d" , &x);
6:     if( x < 1000)
7:     {
8:         a = x%10;
9:         b = ( x/10) % 10;
10:        c = x/100;
11:        printf( " %d, %d, %d" , a, b , c);
12:    }
13:    else
14:        printf( " x >= 1000" );
15: }
```

回答下列问题:

(1) 第 4 行的作用是什么?

(2) 第 8 行, 如果  $x = 678$ , 运行此行语句后  $a =$  \_\_\_\_\_。

(3) 第 9 行, 如果  $x = 678$ , 运行此行语句后  $b =$  \_\_\_\_\_。

(4) 第 10 行, 如果  $x = 678$ , 运行此行语句后  $c =$  \_\_\_\_\_。

(5) 运行结果测试:

①输入 678, 输出 \_\_\_\_\_;

②输入 25, 输出 \_\_\_\_\_;

③输入 1234, 输出 \_\_\_\_\_。

3. 数据类型的自动转换与强制转换。分析下面的程序, 并回答问题。

```
1: main( )
2: {
3:     int m, x;
4:     float fl, f2 = 1.23456789e12;
5:     char c = 268;
```

```

6:     unsigned int y = 65535;
7:     m = 7.68;
8:     f1 = (float) m;
9:     x = y;
10:    printf( " \nm = %d, f1 = %f, f2 = %f" , m, f1, f2) ;
11:    printf( " \nc = %c, y = %u, x = %d" , c, y, x) ;
12:    }

```

回答下列问题:

- (1) 第4行, f2 的值是\_\_\_\_\_。
- (2) 第5行, c 的值是\_\_\_\_\_。
- (3) 第7行, m 的值是\_\_\_\_\_, 它是自动类型转换还是强制类型转换?
- (4) 第8行, f1 的值是\_\_\_\_\_, 它是自动类型转换还是强制类型转换?
- (5) 第9行, x 的值是\_\_\_\_\_, 说明理由。
- (6) 输出结果: \_\_\_\_\_。

4. 自加、自减的前缀、后缀运算。分析程序运行结果, 并回答问题。

```

1:  # include <stdio. h >
2:  void main( )
3:  {   int i, j, k;
4:      int m, n, p;
5:      i = 8; j = 10; k = 12;
6:      m = ++i;
7:      printf( " i = %d\n" , i) ;
8:      printf( " m = %d\n" , m) ;
9:      n = j - - ;
10:     printf( " j = %d\n" , j) ;
11:     printf( " n = %d\n" , n) ;
12:  }

```

回答下列问题:

- (1) 第1行语句起什么作用? 是否可以省略?
- (2) 第5行, 此行有几个语句? 是什么语句?
- (3) 运行第6行后, m = \_\_\_\_\_, i = \_\_\_\_\_。
- (4) 第6行可等价于两条语句, 即\_\_\_\_\_。
- (5) 运行第9行后, n = \_\_\_\_\_, j = \_\_\_\_\_。
- (6) 第9行可等价于两条语句, 即\_\_\_\_\_。

5. 逻辑运算与关系运算。分析程序运行结果, 并回答问题。

```

1:  void main( )
2:  {
3:      int a = 1, b = 2, c = 3, logic;
4:      logic = a + b > c && b <= c;

```

```

5:     printf( " logic = % d\n" , logic);
6:     logic = a >=b + c || b ==c;
7:     printf( " logic = % d\n" , logic);
8:     logic =! ( a <c) +b! =1 && ( a + c) /2;
9:     printf( " logic = % d \n" , logic);
10:  }

```

回答下列问题:

(1) 第 4 行, 这里有 5 个运算符, 即 =、+、>、&& 和 <=, 请排出其优先级, 并写出此语句的运算过程 (每次执行 1 个运算)。

(2) 第 4 行, 此语句运算结果的值 logic 是\_\_\_\_\_。

(3) 第 6 行, 有运行符 =、+、>=、||、==, 请排列其优先级, 并将此语句用括号括起来, 使表达式的值不变。

(4) 第 8 行, 这里的第 1 个符号! 是什么运算? 第 2 个符号! 又是什么运算?

(5) 第 8 行, 此语句运算的结果 logic 的值是多少?

#### 6. 编程题

输入一个年份 y, 判断是否是闰年。如果是闰年, 输出 “y is a leap year”; 否则, 输出 “y is not a leap year”。

闰年的确定:

(1) 如果一个年份能被 4 整除, 但不能被 100 整除就是闰年;

(2) 如果一个年份能被 400 整除, 就是闰年;

(3) 其他年份不是闰年。

#### 7. 编程题

从键盘输入 1 个字符, 如果是英文字符, 将它们转换成大写字母并输出, 否则输出字符的 ASCII 码值。

### 【实验评价】

实验任务评价表

题号	完成情况	知识点掌握情况	难度系数	测试评价
1		输出控制格式	1	<input type="checkbox"/> 了解 <input type="checkbox"/> 掌握 <input type="checkbox"/> 应用
2		算术运算	2	<input type="checkbox"/> 了解 <input type="checkbox"/> 掌握 <input type="checkbox"/> 应用
3		数据类型转换	3	<input type="checkbox"/> 了解 <input type="checkbox"/> 掌握 <input type="checkbox"/> 应用
4		自加、自减运算	4	<input type="checkbox"/> 了解 <input type="checkbox"/> 掌握 <input type="checkbox"/> 应用
5		逻辑运算、关系运算	4	<input type="checkbox"/> 了解 <input type="checkbox"/> 掌握 <input type="checkbox"/> 应用
6		综合应用	4	<input type="checkbox"/> 了解 <input type="checkbox"/> 掌握 <input type="checkbox"/> 应用
7		综合应用	4	<input type="checkbox"/> 了解 <input type="checkbox"/> 掌握 <input type="checkbox"/> 应用

注: 难度系数为 1 ~ 5, 其中: 1 为容易, 5 为难。

## 【思考与讨论】

1. 运算符的综合应用。

(1) 假设有定义 `int a = 7; float x = 2.5, y = 4.7;`, 则表达式 `x + a%3 * 3 * (int)(x + y) % 2 / 4` 的值是 ( )。

A. 2.500000      B. 2.175000      C. 3.150000      D. 0.100000

(2) 已知 `int a = 1, b = 2, c = 3; c = c >= a + b;`, 则 `c` 的值是 ( )。

A. 0      B. 1      C. 3      D. 4

(3) 假设 `int a = 66, b; char c = 'A'; b = (a > c && c < 'a' || !c);`, 则 `b` 的值是 ( )。

A. 0      B. 1      C. 'A'      D. 不确定

(4) 假设 `int a = 1, b = 2; a = a < b ? b : a;`, 执行后 `a`、`b` 的值是 ( )

A. `a = 1, b = 2`      B. `a = 2, b = 1`      C. `a = 1, b = 1`      D. `a = 2, b = 2`

(5) 假设 `x`、`i`、`j` 和 `k` 都是 `int` 型变量, 则执行下面运算后 `x` 值为 ( )。

`x = (i = 4, j = 16, k = 32)`

A. 4      B. 16      C. 32      D. 52

(6) 下列运算符中优先级最高的是 ( )。

A. `<`      B. `+`      C. `&&`      D. `!=`

(7) 假设 `int a, b; float x;`, 则正确的赋值语句是 ( )。

A. `a = 1, b = 2;`      B. `b ++;`      C. `a = b = 5`      D. `b = int(x);`

2. 判断一个数 `m` 能否被另一个数 `n` 整除, 如何用 C 语言表达式描述?

3. `long x = 100000; printf("%d", x);` 输出的结果是多少? 详细说明其转换过程。