

**《C语言程序设计》**

**学习指南**

目录

[一、 课程特点 1](#_Toc22392)

[二、课程目标 1](#_Toc20510)

[三、课程重点难点 1](#_Toc3720)

[四、经典C语言程序设计例子 2](#_Toc30375)

[第二章 数据类型、运算符与表达式、简单程序设计 2](#_Toc16105)

[第三章 顺序程序设计 10](#_Toc25651)

[第四章 选择结构程序设计 15](#_Toc14927)

[第五章 循环控制 16](#_Toc25983)

[第六章 数组 18](#_Toc7442)

[第七章 函数 19](#_Toc17890)

[第九章 指针 20](#_Toc11268)

[第十章 结构体与共用体 21](#_Toc10840)

[五、C语言课程设计题目 24](#_Toc26078)

[题目一：一卡通消费记录管理系统 24](#_Toc30139)

[题目二：个人邮箱系统 25](#_Toc3336)

[题目三：学生考勤信息管理系统 27](#_Toc2014)

[六、教学参考书 28](#_Toc21720)

[七、C语言函数 28](#_Toc30312)

[1、数学函数 28](#_Toc30065)

[2、字符函数 30](#_Toc5535)

[3、字符串函数 31](#_Toc17381)

[4、输入输出函数 31](#_Toc6284)

[5、动态分配函数和随机函数 33](#_Toc6235)

## 课程特点

[C](https://baike.baidu.com/item/C/22337" \t "https://baike.baidu.com/item/c%E8%AF%AD%E8%A8%80/_blank)语言是一门通用[计算机编程语言](https://baike.baidu.com/item/%E8%AE%A1%E7%AE%97%E6%9C%BA%E7%BC%96%E7%A8%8B%E8%AF%AD%E8%A8%80" \t "https://baike.baidu.com/item/c%E8%AF%AD%E8%A8%80/_blank)，广泛应用于底层开发。C语言的设计目标是提供一种能以简易的方式[编译](https://baike.baidu.com/item/%E7%BC%96%E8%AF%91/1258343" \t "https://baike.baidu.com/item/c%E8%AF%AD%E8%A8%80/_blank)、处理低级[存储器](https://baike.baidu.com/item/%E5%AD%98%E5%82%A8%E5%99%A8/1583185" \t "https://baike.baidu.com/item/c%E8%AF%AD%E8%A8%80/_blank)、产生少量的[机器码](https://baike.baidu.com/item/%E6%9C%BA%E5%99%A8%E7%A0%81/86125" \t "https://baike.baidu.com/item/c%E8%AF%AD%E8%A8%80/_blank)以及不需要任何运行环境支持便能运行的编程语言。尽管C语言提供了许多低级处理的功能，但仍然保持着良好跨平台的特性，以一个标准规格写出的C语言程序可在许多电脑平台上进行编译，甚至包含一些嵌入式[处理器](https://baike.baidu.com/item/%E5%A4%84%E7%90%86%E5%99%A8" \t "https://baike.baidu.com/item/c%E8%AF%AD%E8%A8%80/_blank)（单片机或称[MCU](https://baike.baidu.com/item/MCU/62773" \t "https://baike.baidu.com/item/c%E8%AF%AD%E8%A8%80/_blank)）以及超级电脑等作业平台。C语言是一个有结构化程序设计、具有变量作用域以及递归功能的过程式语言。C语言简洁紧凑、灵活方便，运算符丰富，数据类型丰富，表达方式灵活实用，允许直接访问物理地址，对硬件进行操作，生成目标代码质量高，程序执行效率高，可移植性好，表达力强。

## 二、课程目标

本课程是计算机学院各专业必修的专业基础课，课程以课堂教学为主，辅以实验教学，使学生通过本课程的学习后，能全面掌握编程语言所有主要的语言元素，包括各数据类型，流程控制和标准函数库，并熟练地掌握结构化程序设计和变参数的函数功能设计，具备初步的高级语言程序设计能力。

**应达到的要求**

1、了解面向过程的程序设计技术和方法，掌握C语言的基本语法、基本语句、基本控制结构以及自顶向下结构化程序设计的基本方法，具备初步的高级语言程序设计能力，能够意识到算法、良好的程序、设计风格以及实践在本课程学习中的重要性。

2、具备一定的软件开发能力，能够熟练使用编程语言，针对具体问题进行分析和编码，用编程语言设计程序解决实际问题。

3、具备熟练应用编程语言集成环境和开发工具进行程序的编写、编译与调试的能力，具备一定的自学能力和能够利用计算机网络获取新知识、新技术的能力，为将来从事软件研制开发打下坚实的基础。

## 三、课程重点难点

|  |  |  |
| --- | --- | --- |
| 章节名称 | 重点 | 难点 |
| C语言概述和算法初步 | 理解编程语言的特点；初步掌握在计算机上运行程序的方法；理解算法的概念和特性 | 能够用N-S流程图表示算法 |
| 数据类型、运算符与表达式、简单程序设计 | 掌握基本数据类型的表示方法、运算符和表达式概念 | 在掌握基本数据类型的表示方法、运算符和表达式概念的基础上，能够进行简单的程序编写。 |
| 顺序程序设计 | 理解基本的语句，如控制语句、赋值语句等 | 能够结合输入输出语句设计编写简单的程序 |
| 选择结构程序设计 | 了解关系运算符和关系表达式，逻辑运算符和逻辑表达式 | 在了解关系运算符和关系表达式，逻辑运算符和逻辑表达式的基础上，能够编写具有选择结构的程序 |
| 循环控制 | 能够区别多种循环，具备实现循环的能力 | 编写具有循环嵌套的程序 |
| 数组 | 能够定义和引用一维数组、二维数组和字符数组 | 能够利用数组来设计相关程序 |
| 函数 | 掌握函数的定义和调用方法、变量的作用域和存储方式等概念；理解函数调用时参数传递的过程 | 能够区分值传递方式和地址传递；能够用函数实现模块化的程序设计 |
| 指针 | 能够理解和使用指针变量 | 能够设计通过指针引用数组和字符串以及指向函数的指针的程序 |
| 结构体与共用体 | 能够定义和使用结构体类型变量；能够定义和使用结构体数组 | 初步掌握用指针处理链表的方法；了解共用体和枚举类型的概念 |

## 四、经典C语言程序设计例子

### 第二章 数据类型、运算符与表达式、简单程序设计

1. 用printf语句显示多个字符串

编写程序，用printf语句将3个字符串"Let's"，

"study"和"programming"用空格分隔显示在同

一行。

【程序】

void main()

{

printf("Let's "); //末尾加一个空格，但不换行

printf("study "); //末尾加一个空格，但不换行

printf("programming\n"); //显示后换行

}

2. 显示字符串

编写程序，显示字符串"Let's study C."并换行。

【程序】

void main()

{

printf("Let's study C.\n"); //'\n'为换行符，可出现在字符串中

}

3. 用printf语句显示多个字符串

编写程序，用printf语句将3个字符串"Let's"，

"study"和"programming"用空格分隔显示在同

一行。

【程序】

void main()

{

printf("Let's "); //末尾加一个空格，但不换行

printf("study "); //末尾加一个空格，但不换行

printf("programming\n"); //显示后换行

}

4. 换行显示

编写程序，在显示字符串"C Language"后，换行显示字符串"C++ Language"。

【程序】

void main()

{

printf("C Language\n"); //显示后换行

printf("C++ Language\n"); //显示后换行

}

5. 整型变量，赋值，四则运算

编写程序，分别给变量ａ赋值15和变量ｂ赋值3，

然后依次求并显示两者的和、差、积、商。

【程序】

void main()

{

int a,b,sum,min,tim,sep; //整形变量说明

a=15; //赋值

b=3; //赋值

sum=a+b; //加

min=a-b; //减

tim=a\*b; //乘

sep=a/b; //除

printf("%d+%d=%d\n",a,b,sum);

printf("%d-%d=%d\n",a,b,min);

printf("%d\*%d=%d\n",a,b,tim);

printf("%d/%d=%d\n",a,b,sep);

}

6. 余数

编写程序，将215÷9的商和余数分别赋给c和d。

【程序】

void main()

{

int a,b,c,d; //整形变量说明

a=215; //赋值

b=9; //赋值

c=a/b; //将商赋给ｃ

d=a%b; //将余数赋给ｄ

printf("%d/%d=%d...%d\n",a,b,c,d); //显示

}

7. 算术表达式

编写程序，不使用变量，显示35\*25的值。

【程序】

void main()

{

printf("%d\*%d=%d\n",35,25,35\*25);//显示结果

}

8. 算术表达式

编写程序，先对整型变量a赋值15，b赋值10，然后将

a+b与a-b的商赋给c，并显示c的值。

【程序】

void main()

{

int a,b,c;

a=15;

b=10;

c=(a+b)/(a-b);

printf("(%d+%d)/(%d-%d)=%d\n",a,b,a,b,c);//显示结果

}

9. 逗号运算符（也称为顺序运算符）

编写程序，求a\*b与c\*d之和，其中用顺序运算符对

变量a，b，c和d依次赋值为8，7，5和6。

【程序】

void main()

{

int a,b,c,d,timsum;

a=8,b=7,c=5,d=6; //使用逗号分隔，4个赋值构成一个语句

timsum=a\*b+c\*d;

printf("%d\*%d+%d\*%d=%d\n",a,b,c,d,timsum);

}

10. 重复赋值

编写程序，对变量a和b均赋值为5，变量c赋值为8，

分行显示a，b和c的平方值。

【程序】

void main()

{

int a,b,c;

a=b=5,c=8; //对a和b重复赋值

printf("%d\*%d=%d\n",a,a,a\*a);

printf("%d\*%d=%d\n",b,b,b\*b);

printf("%d\*%d=%d\n",c,c,c\*c);

}

11. 变量初始化

编写程序，变量a和b分别赋值为123和20，求a和b的乘积。

要求在进行变量说明的同时给变量赋值。

【程序】

void main()

{

int a=123,b=20; //在说明变量a和b的同时对其赋值(初始化)

int c;

c=a\*b;

printf("%d\*%d=%d\n",a,b,c);

}

12. 浮点数（实数）

编写程序，将789.456123赋给变量a，28.2468赋给变量b，

求a和b的和、差、积、商。

【程序】

void main()

{

float a,b,sum,min,tim,sep; //实数型变量说明

a=789.456123;

b=28.2468;

sum=a+b;

min=a-b;

tim=a\*b;

sep=a/b;

printf("%f+%f=%f\n",a,b,sum);

printf("%f-%f=%f\n",a,b,min);

printf("%f\*%f=%f\n",a,b,tim);

printf("%f/%f=%f\n",a,b,sep);

}

13. 浮点数计算

编写程序，将圆周率3.14159赋给变量a，半径7.24赋给变量b，

求其圆面积和圆周长。

【程序】

void main()

{

float a,b,c,d;

a=3.14159;

b=7.24;

c=b\*b\*a;

d=2\*a\*b;

printf("area=%.3f\n",c); //按3位小数的格式显示结果

printf("circ=%.4f\n",d); //按4位小数的格式显示结果

}

14. 将整数转换为浮点数

编写程序，对变量a赋值为321，b赋值为123.4567，求其浮点数的和。

【程序】

void main()

{

int a;

float b,c;

a=321,b=123.4567;

c=(float)a+b;

printf("%f\n",c);

}

15. 浮点数转换为整数

编写程序，令变量b取值35.425，c取值52.954，

先将b+c的结果转换为整数后赋给变量a1，然后

将b和c的值分别转换为整数后求和赋给变量a2。

【程序】

void main()

{

int a1,a2;

float b,c;

b=35.425,c=52.954;

a1=(int)(b+c);

a2=(int)b+(int)c;

printf("%d\n",a1);

printf("%d\n",a2);

}

16. 复合的赋值运算符

编写程序，首先令变量a值为2，b值为5，c值为6，d值为10，

然后依次将a加b的值赋予a，b减c的值赋予b，c乘以d的值赋

予c，d除以a的值赋予d，a除以c的余数赋予a。

【程序】

void main()

{

int a=2,b=5,c=6,d=10;

a+=b;

b-=c;

c\*=d;

d/=a;

a%=c;

printf("%d %d %d %d %d\n",a,b,c,d,a);

//用空格分隔，在同一行显示各运算结果

}

17. 增1运算：++

编写程序，先对整型变量a赋值为80，然后依次分行显示

++a，a++和a的值。

【程序】

void main()

{

int a=80;

printf("%d\n",++a); //前++

printf("%d\n",a++); //后++

printf("%d\n",a);

}

18. 减1运算：--

编写程序，先令整型变量a的值为80，然后依次分行

显示--a，a--和a的值。

【程序】

void main()

{

int a=80;

printf("%d\n",--a);

printf("%d\n",a--);

printf("%d\n",a);

}

19. 移位运算：左移 <<，右移 >>

编写程序，先令整型变量a值为7，b值为11，依次将a

的值右移1位赋给c，将b的值左移3位赋给d，最后显示

c和d的值。

【程序】

void main()

{

int a=7,b=11;

int c,d;

c=a>>1; //c赋值为a的值右移1个二进位的结果

d=b<<3; //d赋值为b的值左移3个二进位的结果

printf("%d %d\n",c,d);

}

20. 运算符&，|和^的组合运用

编写程序，先令变量a取值5，b取值12，c取值4，

求a|b和a^c的结果的&。

【程序】

void main()

{

int a=5,b=12,c=4;

int er,ok;

er=a&b|a^c; //不符合题意

ok=(a|b)&(a^c); //注意运算的优先级和括号的使用

printf("%d|%d&%d^%d=%d error\n",a,b,a,c,er);

printf("(%d|%d)&(%d^%d)=%d ok\n",a,b,a,c,ok);

}

21. 按位取反运算：~

编写程序，先令整型变量a取值9，然后对a的值

按位取反后赋给b，显示b的值。

【程序】

void main()

{

int a=9;

int b;

b=~a;

printf("%d\n",b);

}

22. 字符型变量

编写程序，先将字符'C'赋予变量a、'H'赋予b，字符'N'赋予c，

然后在同一行用空格分隔显示a、b、c的值。

【程序】

void main\_ans()

{

char a,b,c;

a='C';

b='H';

c='N';

printf("%c %c %c\n",a,b,c);

}

/\*输出：

C H N

\*/

23. 字符显示

编写程序，先将字符'A'、'n'、'y'、'v'、'i'、'e'、'w'分别赋

给变量a、b、c、d、e、f、g，然后在同一行依次显示各变量的值。

【程序】

void main()

{

char a,b,c,d,e,f,g;

a='A',b='n',c='y',d='v';

e='i',f='e',g='w';

printf("%c%c%c%c%c%c%c\n",a,b,c,d,e,f,g);

}

/\*显示：

Anyview

\*/

24. 字符和ASCII码

编写程序，先将字符'C'、'H'、'N'分别赋给变量a，b，c，

然后显示变量a、b、c的字符值和ASCII码。

【程序】

void main()

{

char a,b,c;

a='C',b='H',c='N';

printf("%c:%d %c:%d %c:%d\n",a,a,b,b,c,c);

}

/\*显示：

C:67 H:72 N:78

\*/

25. ASCII码和字符

编写程序，先将67，72，78分别赋给整型变量a，b，c，

然后显示三个变量值及其作为ASCII码所对应的字符。

【程序】

void main()

{

int a,b,c;

a=67,b=72,c=78;

printf("%d:%c %d:%c %d:%c\n",a,a,b,b,c,c);

}

/\*显示：

67:C 72:H 78:N

\*/

### 第三章 顺序程序设计

1. 用scanf输入

编写程序，用scanf输入2个整数到变量a和b，求其之积并显示等式。

【程序】

void main()

{

int a,b,c;

scanf("%d",&a);

scanf("%d",&b);

c=a\*b;

printf("%d\*%d=%d\n",a,b,c);

}

2. 用一个scanf语句输入两个整数

编写程序，用一个scanf语句输入两个整数，求其和并显示等式。

【程序】

void main()

{

int a,b,c;

scanf("%d %d",&a,&b);

c=a+b;

printf("%d+%d=%d\n",a,b,c);

}

3. 用一个scanf语句输入两个实数

编写程序，用一个scanf语句输入两个实数，求其差并显示等式。

【程序】

void main()

{

float a,b,c;

scanf("%f %f",&a,&b);

c=a-b;

printf("%f-%f=%f\n",a,b,c);

}

4. 用scanf语句输入一个字符

编写程序，用scanf语句输入1个字符到字符变量a，显示其字符值和ASCII码。

【程序】

void main()

{

char a;

scanf("%c",&a);

printf("%c:%d\n",a,a);

}

5. 用一个scanf语句输入两个字符

编写程序，用一个scanf语句输入两个字符到字符变量a和b，

显示其字符值和ASCII码。

【程序】

void main()

{

char a,b;

scanf("%c%c",&a,&b);

printf("%c:%d %c:%d\n",a,a,b,b);

}

6. 用一个scanf语句输入两个字符（忽略分隔空格）

编写程序，用一个scanf语句,忽略分隔空格，输入两个字符到字符

变量a和b，显示其字符值和ASCII码。

【程序】

void main()

{

char a,b;

scanf("%c %c",&a,&b);

printf("%c:%d %c:%d\n",a,a,b,b);

}

7. 用两个scanf语句各输入一个字符

编写程序，用两个scanf语句各输入一个字符到字符

变量a和b，显示其字符值和ASCII码。

【程序】

void main( )

{

char a,b;

scanf("%c",&a);

scanf("%c",&b);

printf("%c:%d %c:%d\n",a,a,b,b);

}

8. 忽略中间一个字符的输入

编写程序，用两个scanf语句，忽略中间一个字符，分别输入

一个字符到字符变量a和b，显示其字符值和ASCII码。

【程序】

void main()

{

char a,b;

scanf("%c",&a);

scanf("%\*c%c",&b);

printf("%c:%d %c:%d\n",a,a,b,b);

}

9. 16进制数和8进制数的输出

编写程序，显示175的16进制表示和23的8进制表示。

【程序】

void main()

{

int a,b;

a=175;

b=23;

printf("a=(%d)10#=(%x)16#\n",a,a);

printf("b=(%d)10#=(%o)8#\n",b,b);

}

10. 科学记数法

编写程序，分别赋值变量a为123.4567和b为97531.68，

并用科学记数法形式显示。

【程序】

void main()

{

float a,b;

a=123.4567;

b=97531.68;

printf("a=%f=%e\n",a,a); /\*a=123.456700=1.234567e+002\*/

printf("b=%f=%e\n",b,b); /\*b=97531.680000=9.753168e+004\*/

}

11. 输入和显示16进制数

编写程序，用scanf语句输入一个16进制数到变量a，

并显示其16进制和10进制表示。

【程序】

void main()

{

int a;

scanf("%x",&a);

printf("a=(%x)16#=(%d)10#\n",a,a);

}

12. 输入和显示8进制数

编写程序，用一个scanf语句输入两个8进制数

到变量a和b，显示a和b的乘积。

【程序】

void main()

{

int a,b;

scanf("%o %o",&a,&b);

printf("%o\*%o=%o\n",a,b,a\*b);

}

13. long长整型变量

编写程序，将123456789和87654321分别赋给

长整型变量a和b，求其和赋给sum并显示结果。

【程序】

void main()

{

long a,b,sum;

a=123456789;

b=87654321;

sum=a+b;

printf("%ld+%ld=%ld\n",a,b,sum);

}

14. unsigned int无符号整型变量

编写程序，将88、89和90分别赋给无符号整型变量

a、b和c，显示其值和相应字符。

【程序】

void main()

{

unsigned int a,b,c;

a=88;

b=89;

c=90;

printf("%d:%c %d:%c %d:%c\n",a,a,b,b,c,c);

}

15. 指定显示位数

编写程序，将135、43和6789依次赋给整型变量

a、b和c，分别用域宽5位、3位、9位进行显示。

【程序】

void main()

{

int a,b,c;

a=135,b=43,c=6789;

printf("123456789\n");

printf("%5d\n",a);

printf("%3d\n",b);

printf("%9d\n",c);

}

16. 指定浮点数的显示格式

编写程序，将543.3456和23.4567赋给浮点型变量

a和b，求其和、差、积、商，并在显示时分别指定

了不同的域宽和小数点后的位数。

【程序】

void main()

{

float a,b,sum,min,tim,sep;

a=543.3456;

b=23.4567;

sum=a+b;

min=a-b;

tim=a\*b;

sep=a/b;

printf("%10.4f+%10.4f=%10.4f\n",a,b,sum);

printf("%10.2f-%10.2f=%10.2f\n",a,b,min);

printf("%10.5f\*%10.5f=%10.5f\n",a,b,tim);

printf("%10.4f/%10.4f=%5.2f\n",a,b,sep);

}

17. 指定小数点后的显示位数

编写程序，输入一个浮点数，分别按小数点后

3位、2位和1位的格式显示。

【程序】

void main()

{

float a;

scanf("%f",&a);

printf("%.3f %.2f %.1f\n",a,a,a);

}

18. 显示字符图案

编写程序，显示字符图案:

x x

x x

xx

xx

x x

x x

【程序】

void main()

{

char a;

a='x';

printf("%1c%5c\n",a,a);

printf("%2c%3c\n",a,a);

printf("%3c%1c\n",a,a);

printf("%3c%1c\n",a,a);

printf("%2c%3c\n",a,a);

printf("%1c%5c\n",a,a);

}

19. 指定有效输入位数

编写程序，分别按1位和2位输入两个整数到整型

变量a和b，求其积赋给c并显示。

【程序】

void main()

{

int a,b,c;

scanf("%1d%2d",&a,&b);

c=a\*b;

printf("%d\*%d=%d\n",a,b,c);

}

### 第四章 选择结构程序设计

1. 输入三个整数 x,y,z，请把这三个数由小到大输出。

程序分析： 我们想办法把最小的数放到 x 上，先将 x 与 y 进行比较， 如果 x>y 则将 x 与 y 的值进行交换，然后再用 x 与 z 进行比较，如果 x>z 则将 x 与 z 的值进行交换，这样能使 x 最小。

程序源代码：

int main()

{

int x,y,z,t;

scanf("%d%d%d",&x,&y,&z);

if (x>y)

{t=x;x=y;y=t;} /\*交换 x,y 的值 \*/

if(x>z)

{t=z;z=x;x=t;}/\* 交换 x,z 的值 \*/

if(y>z)

{t=y;y=z;z=t;}/\* 交换 z,y 的值 \*/

printf("small to big: %d %d %d\n",x,y,z);

return 0;

}

2. 利用条件运算符的嵌套来完成此题：学习成绩 >=90 分的同学用 A 表示， 60-89 分之间的用 B 表示，60 分以下的用 C 表示。

程序分析： (a>b)?a:b 这是条件运算符的基本例子。

程序源代码：

int main()

{

int score;

char grade;

printf("please input a score\n");

scanf("%d",&score);

grade=score>=90?'A':(score>=60?'B':'C');

printf("%d belongs to %c",score,grade);

return 0;

}

### 第五章 循环控制

1. 有 1、2、3、4 个数字，能组成多少个互不相同且无重复数字的三位数？都是多少？

程序分析：可填在百位、十位、个位的数字都是 1、 2、3、4。组成所有的排列后再去掉不满足条件的排列。

程序源代码：

int main()

{

int i,j,k;

printf("\n");

for(i=1;i<5;i++) ／\*以下为三重循环 \*/

for(j=1;j<5;j++)

for (k=1;k<5;k++)

{

if (i!=k&&i!=j&&j!=k) /\*确保 i、j、k 三位互不相同 \*/

printf("%d,%d,%d\n",i,j,k);

}

return 0;

}

2. 打印出所有的“水仙花数” ，所谓“水仙花数”是指一个三位数，其各位数字立方和等于该数本身。

例如： 153 是一个“水仙花数” ，因为 153=1 的三次方＋ 5的三次方＋ 3 的三次方。

程序分析：利用 for 循环控制 100-999 个数，每个数分解出个位，十位，百位。

程序源代码：

int main()

{

int i,j,k,n;

printf("'water flower'number is:");

for(n=100;n<1000;n++)

{

i=n/100;/\* 分解出百位 \*/

j=n/10%10;/\* 分解出十位 \*/

k=n%10;/\* 分解出个位 \*/

if(i\*100+j\*10+k==i\*i\*i+j\*j\*j+k\*k\*k)

{

printf("%-5d",n);

}

}

printf("\n");

}

3. 题目：猴子吃桃问题：猴子第一天摘下若干个桃子，当即吃了一半，还不瘾，又多吃了一个，第二天早上又将剩下的桃子吃掉一半，又多吃了一个。以后每天早上都吃了前一天剩下的一半零一个。到第 10 天早上想再吃时，见只剩下一个桃子了。求第一天共摘了多少。

程序分析：采取逆向思维的方法，从后往前推断。

程序源代码：

int main()

{

int day,x1,x2;

day=9;

x2=1;

while(day>0)

{

x1=(x2+1)\*2;/\* 第一天的桃子数是第 2 天桃子数加 1 后的 2 倍 \*/

x2=x1;

day--;

}

printf("the total is %d\n",x1);

return 0;

}

4. 题目：打印出如下图案（菱形）

\*

\*\*\*

\*\*\*\*\*\*

\*\*\*\*\*\*\*\*

\*\*\*\*\*\*

\*\*\*

\*

程序分析：先把图形分成两部分来看待，前四行一个规律，后三行一个规律，利用双重 for 循环，第一层控制行，第二层控制列。

程序源代码：

int main()

{

int i,j,k;

for(i=0;i<=3;i++)

{

for(j=0;j<=2-i;j++)

printf(" ");

for(k=0;k<=2\*i;k++)

printf("\*");

printf("\n");

}

for(i=0;i<=2;i++)

{

for(j=0;j<=i;j++)

printf(" ");

for(k=0;k<=4-2\*i;k++)

printf("\*");

printf("\n");

}

return 0;

}

### 第六章 数组

1. 求一个 3\*3 矩阵对角线元素之和

程序分析：利用双重 for 循环控制输入二维数组，再将 a[i][i] 累加后输出。

程序源代码：

int main()

{

float a[3][3],sum=0;

int i,j;

printf("please input rectangle element:\n");

for(i=0;i<3;i++)

for(j=0;j<3;j++)

scanf("%f",&a[i][j]);

for(i=0;i<3;i++)

sum=sum+a[i][i];

printf("duijiaoxian he is %6.2f",sum);

}

2. 将一个数组逆序输出。

程序分析：用第一个与最后一个交换。

程序源代码：

#define N 5

int main()

{

int a[N]={9,6,5,4,1},i,temp;

printf("\n original array:\n");

for(i=0;i<N;i++)

printf("%4d",a[i]);

for(i=0;i<N/2;i++)

{

temp=a[i];

a[i]=a[N-i-1];

a[N-i-1]=temp;

}

printf("\n sorted array:\n");

for(i=0;i<N;i++)

printf("%4d",a[i]);

return 0;

}

### 第七章 函数

1. 写一函数求 3 个整数中最小的数。

int min(int x, int y, int z)

/\* 返回 3 个整数 x,y 和 z 中最小的数 \*/

{

if(x>y)

x=y;

if(x>z)

x=z;

return x;

}

2. 先编写一个判断素数的函数。再编写一个函数

将一个偶数表示为两个素数之和，并返回其中较小的素数。

注：素数指只能被 1 和自身整除的正整数。规定 0，1 不是素数。

int prime(int n)

/\* 判断素数，如果是素数返回 1，不是素数则返回 0 \*/

{

int t;

if(n==1)

return 0;

for(t=2;t<=(n/2);t++)

if(n%t==0)

return 0;

return 1;

}

int f(int i)

/\* 将偶数 i 表示为两个素数之和，返回其中较小的素数 \*/

{

int n;

for(n=3;n<=i;n++)

if(prime(i-n)&&prime(n))

return n;

}

3. 编写函数将一个 nxn 的二维数组按“次对角线”翻转。

例如：翻转前的数组 翻转后的数组

1 2 3 9 6 3

4 5 6 ---> 8 5 2

7 8 9 7 4 1

void invert(char a[N][N])

{

int i,j;

int b[N][N];

for(i=0;i<N;i++)

for(j=0;j<N;j++)

b[i][j]=a[N-1-j][N-i-1];

for(i=0;i<N;i++)

for(j=0;j<N;j++)

a[i][j]=b[i][j];

}

### 第九章 指针

1. 编写函数，通过指针参数 sum 将 x 和 y 的

和赋给相应实际参数。例如，若 x=134、y=378，则 sum

的实参变量被赋值 512。

void func(int x, int y, int \*sum)

{

\*sum=x+y;

}

2. 编写函数，比较指针参数 x 和 y 分别

指向的变量，若前者的值较大，则交换两者的值，

并返回 1；否则无须交换，返回 0。

int func(int \*x, int \*y)

{

int m;

if(\*x>\*y)

{

m=\*x;

\*x=\*y;

\*y=m;

return 1;

}

else

return 0;

}

3. 编写函数，查找并返回 p 指示的二维数

组中的元素最大值。例如，二维数组的元素值依次为

{{45,38,-6,78},{34,48,70,4},{-9,0,4,20}} ，则

函数返回值为 78。

int func(int (\*p)[4], int n)

{

int i ,j,min=p[0][0];

for(i=0;i<n;i++)

for(j=0;j<4;j++)

if(p[i][j]>min)

min=p[i][j];

return min;

}

### 第十章 结构体与共用体

1. 结构体类型定义如下：

struct date{int year; int month; int day;}; // 定义日期结构体类型

struct student

{

char id[10]; //学号

char name[20]; //姓名

struct date birth; //出生日期

};

结构体数组 s 存储了 n 个人的学号、名字和出生日期。写一函数，以结构体的形式

返回这 n 个人中年龄最大（即出生日期最小）者的信息。

struct student oldest(struct student s[], int n)

{

int i;

struct student \*p;

p=&s[0];

for(i=1;i<n;i++)

{

if(s[i].birth.year<(\*p).birth.year)

{

p=&s[i]; continue;

}

if(s[i].birth.year==(\*p).birth.year)

{

if(s[i].birth.month<(\*p).birth.month)

{

p=&s[i]; continue;}

if(s[i].birth.month==(\*p).birth.month)

{

if(s[i].birth.day<(\*p).birth.day)

p=&s[i];

}

}

}

return \*p;

}

1. 课程链表结点的结构体类型定义如下：

struct courseNode //课程链表结点的结构体类型

{

int cID; //课程号，取值 0~99

char name[10]; //课程名

float credit; //学分，取值 0~5

int semester; //学期，取值 1~8

struct courseNode \*next;

};

结构体链表 Lc 存储了多门课程的信息。写一函数，将课程号为 c 的

课程结点删除。

struct courseNode \*deleteCourse(struct courseNode \*\*Lc, int c)

/\* 若在链表 Lc 中课程 c 不存在，则删除不成功，返回 null;

否则从链表 Lc 中删除该课程结点，并返回指向该课程结点的指针。

\*/

{

struct courseNode \*p1,\*p2,\*head,\*t;

p1=p2=\*Lc;

while(c!=p1->cID&&p1->next!=NULL)

{

p2=p1;p1=p1->next;

}

if(c==p1->cID)

{

t=p1;

if(p1==\*Lc)

\*Lc=p1->next;

else

p2->next=p1->next;

return(t);

}

else

return(NULL);

}

1. 单向链表的结点类型定义如下：

struct node{

char ch;

struct node \*next;

};

编写函数，对单向链表 L 实现就地逆置，即将所有结点

的指针反向，原链头当作链尾，原链尾当作链头，并返

回逆置后链表的头指针。

struct node \*inverse(struct node \*L)

{

struct node \*p1=L,\*p2;

int i,n=0;

char t;

while(p1!=NULL)

{

p1=p1->next;

n++;

}

p1=L;

for(i=0;i<n-1;i++)

{

while(p1!=NULL)

{

p2=p1->next;

if((p1->ch)<(p2->ch))

{t=p1->ch,p1->ch=p2->ch,p2->ch=t;}

if(p2->next!=NULL)

p1=p1->next;

else p1=NULL;

}

p1=L;

}

return L;

}

## 五、C语言课程设计题目

### 题目一：一卡通消费记录管理系统

1. 功能需求说明(必须采用结构体和动态链表实现)

1. 消费记录存在文件fee.txt中,格式如下：

每一条记录包括一个消费的交易日期、入账日期、交易额、交易后余额、交易类型、交易次数、交易地点。示例如下



1. 查询搜索功能

（1）用户能够查询自己的一卡通固定时段消费情况：可以按月、学期、学年查询

（2）用户可以自定义一个时间段进行查询自己的消费情况



（3）用户可以按交易金额查找消费记录

（4）管理员可查询任一用户的消费记录，可按照固定时段、自定义时段、金额方式、已删除记录查询

3. 排序功能：按消费记录的交易日期、金额、次数中一个进行（升序或降序）排序。

4. 数据录入：管理员能录入某个用户的一卡通消费情况，可以一次完成若干条记录的输入。

5. 显示功能：可进行全部消费记录的显示，也可按照查询结果进行显示

6. 数据变更：管理员可以更改或删除用户的某条消费记录，但删除的信息要保存在另外一个文件fee\_undel.dat

7. 统计功能：（1）用户可统计自己在一个自定义时间段里的消费总金额情况

（2）管理员可统计各个用户在一个自定义时间段里的消费总金额情况

8. 用户和管理员的权限要分开。

1. 加分项
2. 使用图形界面。
3. 使用线、图表示查询记录和统计消费结果
4. 可参考实际情况对软件进行功能扩充

三、 设计要求

1. 校园卡消费记录文件为 fee.txt，所有消费记录以交易日期从小到大排序（该文件自行建立）。要求编程序实现查询、排序、统计、录入、更新等功能。具体要求如下：

（1）要求显示如下界面

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1------查询 2------排序 3------统计 4------录入 5------更新

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

通过选择 1-4 来确定要做哪一个操作。

（2）若选择 1 则出现如下界面

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1-----按固定时段查询 2-----按日期区间段查询 3-----按交易金额查询

4-----已删除记录查询（管理员才有此项功能）

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

通过选择 1-4 来确定操作,有符合的记录输出到屏幕，若无，则输出相关提示信息。

1. 若选 2，按消费记录的交易日期、金额、次数中一个进行（升序或降序）排序。
2. 若选 3，统计一个时间段里的消费总金额统计，并显示结果。
3. 若选 4，删除指定的消费记录，并把删除后剩下的数据保存到文件 fee\_del.txt 中。
4. 不同的模块都要有出错处理，并能给出出错提示。如输入数据错误，文件操作错误等等
5. 以上各个功能均编写成子函数，有良好的注释说明，由主函数调用实现。
6. 必须采用结构体和动态链表实现对消费记录数据的存储和访问；
7. 必须使用文件保存数据
8. 应提供一个界面来调用各个功能,调用界面和各个功能的操作界面应尽可能清晰美观二、测试数据要求

(1)不少于 3 个学生的消费记录；(2)不少于 10 个月的消费记录；

(3)每个月的消费不少于 5 个； (4)删除的消费记录不少于 20 条；

### 题目二：个人邮箱系统

在结构体数组的基础上，设计要求如下：

个人邮箱系统主要包含是两个子模块：（1）通讯录（2）邮件管理

一、通讯录

该功能根据用户对联系人的分组规划联系人的联系方式，存储用户联系人的联系方式，并对联系方式和分组情况进行管理。

通讯录具体信息如下

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 姓名 | 邮件地址 | 手机号码 | 分组 |
| 1 | 张三丰 | Zsf@qq.com | 13612345678 | 朋友 |

分组信息如下

|  |  |
| --- | --- |
| 序号 | 组名 |
| 1 | 最近联系 |
| 2 | 朋友 |
| … | … |

1. 添加通讯录：根据类别对登陆用户的关系人进行区分，添加新联系人的通讯方式，用户录入联系人的具体信息。提交后存储入数据文件后，自动生成该联系人通讯方式的编号，存储成功后，显示该类别所有联系人的通讯方式。

2.查看通讯录：用户选择查询的条件比如：用户名，邮件地址，手机号码，分组等条件，系统返回和用户输入条件相似的结果。

3.修改通讯录：用户选择要修改的联系人的通讯方式，点击查看详细信息，进行修改提交后，更新数据文件中的数据更新成功后，显示当前联系人的通讯方式详细信息。

4.删除通讯录：用户选择要删除的联系人，点击删除，数据文件中的数据删除成功后，返回选择系统用户的页面。

5.类别管理

（1）添加分组类别：用户录入类别名称，提交存储入数据文件，自动生成类别编号存储成功后显示所有类别。

（2）检索类别：用户查看所有的分类。

（3）修改类别名称：管理员选择要修改的类别，点击修改，显示类别名称修改，提交后数据文件后，更新成功后，显示所有的分类。

（4）删除类别：管理员选择要删除的类别，点击删除，该类别下的所有用户的通讯方式都自动转移到默认的分组（朋友）中。数据删除数据成功后，显示所有的分类。默认分组不可删除。

二、邮件管理

该模块提供对邮件的管理功能，可以草稿箱、接收邮件（收件箱）、发送邮件、删除邮件（垃圾箱）、查看邮件、查找邮件，并可以将已有的邮件程进行显示。详细功能如下

1. 草稿箱：用户可编写新的邮件，或对未编辑完成的邮件进行保存。邮件内容如下。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 邮件标题 | 发送时间 | 内 容 | 发件人 | 收件人 | 状态 |
|  |  | (不超过200个字符) |  |  | 草稿/已发送/已读未读 |

1. 接收邮件（收件箱）：每次登陆系统后可自动接收邮件，也可通过用户手动接收。

3. 发送邮件：用户可对草稿箱中的邮件进行人工发送。

4. 删除邮件（垃圾箱）：用户可对邮件进行删除，删除后的邮件放入垃圾箱。

5. 查看邮件：用户可以对任意邮件进行查看、并可进行编辑、查找、转发功能。

6. 查找邮件：用户选择查询的条件比如：用户名，邮件地址，手机号码，分组等条件。查询邮件，系统返回和用户输入条件相似的结果。

7. 邮件状态转换：可手动将邮件状态进行切换（已读-》未读，未读-》已读）。

三、 设计要求

1. 通讯录数据放入指定文件：Adress.txt,邮件数据放入 mail.txt

2. 不同的模块都要有出错处理，并能给出出错提示。如输入数据错误，文件操作错误等等

3．以上各个功能均编写成子函数，有良好的注释说明，由主函数调用实现。

4．必须采用结构体数组现对通讯录和邮件记录数据的存储和访问，采用链表结构不加分；

5. 必须使用文件保存数据

6. 应提供一个界面来调用各个功能,调用界面和各个功能的操作界面应尽可能清晰美观

四、测试数据要求

（1）通讯录不少于 5 人信息；（2）每人邮箱不少于 20 封邮件；

（3）每人邮箱：草稿箱：不少于 6 封；收件箱：不少于 10 封；垃圾箱：不少于 4 封

题目三：学生考勤信息管理系统

1 建立文件

(1)考勤记录文件使用指定文件名 ARecord.txt；

(2)可以不保存输入记录，但需要确认是否保存输入记录

2.文件的存取和显示

(1)可以单独存取文件；

(2)可以随时显示内存中记录的全部内容；

3 删除记录

(1)可以按“学号”或“姓名”方式删除记录；

(2)能给出被删除的信息，进行删除，如果没有要删除的信息，输出没有找到的信息；

4 查询记录

(1)可以按“姓名”或“学号”方式查询个人信息和考勤登记信息记录

(2)可以按“任课教师”或“课程名称”方式查询课程信息

(3)能给出查询记录的信息；

(4)如果查询的信息不存在，输出没有找到的信息；

5 统计记录

(1)可以显示某生某门课考勤信息，并统计各类违纪情况次数

(2)可以显示某人所有门课考勤信息，并统计各类违纪情况次数

(3)可以显示某班所有课程考勤信息，并统计各类违纪情况次数

6 备份功能

(1) 编写函数 Backupdata 实现将 ARecord.txt 内容复制到备份文件 backup.txt 中，要求不能使用该两个文件以外的任何文件。

7 整体功能

(1)可以随时检索、删除、或增加新记录，保存或取消新的记录

(2)姓名可由 16 位字符和数字的混合编码组成

(3)使电话号码可由 11 位字符和数字组成

(4)将输出信息加上输出信息栏，例如

8 测试数据要求

(1)不少于 3 门课信息（高数、英语、程序设计）。

(2)不少于 1 个班，每班不少于 10 个同学。每人每门课的考勤信息不少于 5 次。

附表 1：数据格式要求

个人信息

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 姓名 | 学号 | 班级 | 专业 | 电话 | 年龄 | 生日 | 宿舍 | QQ |
|  |  |  |  |  |  |  |  |  |

课程信息

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 课程编号 | 课程名称 | 课程学分 | 任课教师 | 上课时间 | 上课次数 | 上课地点 |
|  |  |  |  |  |  |  |

考勤登记

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 姓名 | 学号 | 班级 | 课程编号 | 课程名称 | 考勤违纪 | 第几次课 | 时间 | 地点 |
|  |  |  |  |  | 迟到/早退/旷课/正常 |  |  |  |

## 六、教学参考书

[1] 谭浩强编，《C程序设计（第四版）》，清华大学出版社，2010年出版

[2] 谭浩强，《C程序设计题解与上机指导（第四版）》，清华大学出版社，2010年出版

[3] 谭浩强，《C程序设计试题汇编》，清华大学出版社，2004年出版

## 七、C语言函数

### 1、数学函数

调用数学函数时，要求在源文件中包下以下命令行：

#include <math.h>

|  |  |  |  |
| --- | --- | --- | --- |
| int abs( int x) | 求整数x的绝对值 | 计算结果 |  |
| double fabs(double x) | 求双精度实数x的绝对值 | 计算结果 |  |
| double acos(double x) | 计算cos-1(x)的值 | 计算结果 | x在-1～1范围内 |
| double asin(double x) | 计算sin-1(x)的值 | 计算结果 | x在-1～1范围内 |
| double atan(double x) | 计算tan-1(x)的值 | 计算结果 |  |
| double atan2(double x) | 计算tan-1(x/y)的值 | 计算结果 |  |
| double cos(double x) | 计算cos(x)的值 | 计算结果 | x的单位为弧度 |
| double cosh(double x) | 计算双曲余弦cosh(x)的值 | 计算结果 |  |
| double exp(double x) | 求ex的值 | 计算结果 |  |
| double fabs(double x) | 求双精度实数x的绝对值 | 计算结果 |  |
| double floor(double x) | 求不大于双精度实数x的最大整数 |  |  |
| double fmod(double x,double y) | 求x/y整除后的双精度余数 |  |  |
| double frexp(double val,int \*exp) | 把双精度val分解尾数和以2为底的指数n，即val=x\*2n，n存放在exp所指的变量中 | 返回位数x  0.5≤x<1 |  |
| double log(double x) | 求㏑x | 计算结果 | x>0 |
| double log10(double x) | 求log10x | 计算结果 | x>0 |
| double modf(double val,double \*ip) | 把双精度val分解成整数部分和小数部分，整数部分存放在ip所指的变量中 | 返回小数部分 |  |
| double pow(double x,double y) | 计算xy的值 | 计算结果 |  |
| double sin(double x) | 计算sin(x)的值 | 计算结果 | x的单位为弧度 |
| double sinh(double x) | 计算x的双曲正弦函数sinh(x)的值 | 计算结果 |  |
| double sqrt(double x) | 计算x的开方 | 计算结果 | x≥0 |
| double tan(double x) | 计算tan(x) | 计算结果 |  |
| double tanh(double x) | 计算x的双曲正切函数tanh(x)的值 | 计算结果 |  |

### 2、字符函数

调用字符函数时，要求在源文件中包下以下命令行：

#include <ctype.h>

|  |  |  |
| --- | --- | --- |
| ****函数原型说明**** | ****功能**** | ****返回值**** |
| int isalnum(int ch) | 检查ch是否为字母或数字 | 是，返回1；否则返回0 |
| int isalpha(int ch) | 检查ch是否为字母 | 是，返回1；否则返回0 |
| int iscntrl(int ch) | 检查ch是否为控制字符 | 是，返回1；否则返回0 |
| int isdigit(int ch) | 检查ch是否为数字 | 是，返回1；否则返回0 |
| int isgraph(int ch) | 检查ch是否为ASCII码值在ox21到ox7e的可打印字符（即不包含空格字符） | 是，返回1；否则返回0 |
| int islower(int ch) | 检查ch是否为小写字母 | 是，返回1；否则返回0 |
| int isprint(int ch) | 检查ch是否为包含空格符在内的可打印字符 | 是，返回1；否则返回0 |
| int ispunct(int ch) | 检查ch是否为除了空格、字母、数字之外的可打印字符 | 是，返回1；否则返回0 |
| int isspace(int ch) | 检查ch是否为空格、制表或换行符 | 是，返回1；否则返回0 |
| int isupper(int ch) | 检查ch是否为大写字母 | 是，返回1；否则返回0 |
| int isxdigit(int ch) | 检查ch是否为16进制数 | 是，返回1；否则返回0 |
| int tolower(int ch) | 把ch中的字母转换成小写字母 | 返回对应的小写字母 |
| int toupper(int ch) | 把ch中的字母转换成大写字母 | 返回对应的大写字母 |

### 3、字符串函数

调用字符函数时，要求在源文件中包下以下命令行：

#include <string.h>

|  |  |  |
| --- | --- | --- |
| ****函数原型说明**** | ****功能**** | ****返回值**** |
| char \*strcat(char \*s1,char \*s2) | 把字符串s2接到s1后面 | s1所指地址 |
| char \*strchr(char \*s,int ch) | 在s所指字符串中，找出第一次出现字符ch的位置 | 返回找到的字符的地址，找不到返回NULL |
| int strcmp(char \*s1,char \*s2) | 对s1和s2所指字符串进行比较 | s1<s2,返回负数；s1= =s2,返回0；s1>s2,返回正数 |
| char \*strcpy(char \*s1,char \*s2) | 把s2指向的串复制到s1指向的空间 | s1 所指地址 |
| unsigned strlen(char \*s) | 求字符串s的长度 | 返回串中字符（不计最后的'\0'）个数 |
| char \*strstr(char \*s1,char \*s2) | 在s1所指字符串中，找出字符串s2第一次出现的位置 | 返回找到的字符串的地址，找不到返回NULL |

### 4、输入输出函数

调用字符函数时，要求在源文件中包下以下命令行：

#include <stdio.h>

|  |  |  |
| --- | --- | --- |
| ****函数原型说明**** | ****功能**** | ****返回值**** |
| void clearer(FILE \*fp) | 清除与文件指针fp有关的所有出错信息 | 无 |
| int fclose(FILE \*fp) | 关闭fp所指的文件，释放文件缓冲区 | 出错返回非0，否则返回0 |
| int feof (FILE \*fp) | 检查文件是否结束 | 遇文件结束返回非0，否则返回0 |
| int fgetc (FILE \*fp) | 从fp所指的文件中取得下一个字符 | 出错返回EOF，否则返回所读字符 |
| char \*fgets(char \*buf,int n, FILE \*fp) | 从fp所指的文件中读取一个长度为n-1的字符串，将其存入buf所指存储区 | 返回buf所指地址，若遇文件结束或出错返回NULL |
| FILE \*fopen(char \*filename,char \*mode) | 以mode指定的方式打开名为filename的文件 | 成功，返回文件指针（文件信息区的起始地址），否则返回NULL |
| int fprintf(FILE \*fp, char \*format, args,…) | 把args,…的值以format指定的格式输出到fp指定的文件中 | 实际输出的字符数 |
| int fputc(char ch, FILE \*fp) | 把ch中字符输出到fp指定的文件中 | 成功返回该字符，否则返回EOF |
| int fputs(char \*str, FILE \*fp) | 把str所指字符串输出到fp所指文件 | 成功返回非负整数，否则返回-1（EOF） |
| int fread(char \*pt,unsigned size,unsigned n, FILE \*fp) | 从fp所指文件中读取长度size为n个数据项存到pt所指文件 | 读取的数据项个数 |
| int fscanf (FILE \*fp, char \*format,args,…) | 从fp所指的文件中按format指定的格式把输入数据存入到args,…所指的内存中 | 已输入的数据个数，遇文件结束或出错返回0 |
| int fseek (FILE \*fp,long offer,int base) | 移动fp所指文件的位置指针 | 成功返回当前位置，否则返回非0 |
| long ftell (FILE \*fp) | 求出fp所指文件当前的读写位置 | 读写位置，出错返回 -1L |
| int fwrite(char \*pt,unsigned size,unsigned n, FILE \*fp) | 把pt所指向的n\*size个字节输入到fp所指文件 | 输出的数据项个数 |
| int getc (FILE \*fp) | 从fp所指文件中读取一个字符 | 返回所读字符，若出错或文件结束返回EOF |
| int getchar(void) | 从标准输入设备读取下一个字符 | 返回所读字符，若出错或文件结束返回-1 |
| char \*gets(char \*s) | 从标准设备读取一行字符串放入s所指存储区，用’\0’替换读入的换行符 | 返回s,出错返回NULL |
| int printf(char \*format,args,…) | 把args,…的值以format指定的格式输出到标准输出设备 | 输出字符的个数 |
| int putc (int ch, FILE \*fp) | 同fputc | 同fputc |
| int putchar(char ch) | 把ch输出到标准输出设备 | 返回输出的字符，若出错则返回EOF |
| int puts(char \*str) | 把str所指字符串输出到标准设备，将’\0’转成回车换行符 | 返回换行符，若出错，返回EOF |
| int rename(char \*oldname,char \*newname) | 把oldname所指文件名改为newname所指文件名 | 成功返回0，出错返回-1 |
| void rewind(FILE \*fp) | 将文件位置指针置于文件开头 | 无 |
| int scanf(char \*format,args,…) | 从标准输入设备按format指定的格式把输入数据存入到args,…所指的内存中 | 已输入的数据的个数 |

### 5、动态分配函数和随机函数

调用字符函数时，要求在源文件中包下以下命令行：

#include <stdlib.h>

|  |  |  |
| --- | --- | --- |
| ****函数原型说明**** | ****功能**** | ****返回值**** |
| void \*calloc(unsigned n,unsigned size) | 分配n个数据项的内存空间，每个数据项的大小为size个字节 | 分配内存单元的起始地址；如不成功，返回0 |
| void \*free(void \*p) | 释放p所指的内存区 | 无 |
| void \*malloc(unsigned size) | 分配size个字节的存储空间 | 分配内存空间的地址；如不成功，返回0 |
| void \*realloc(void \*p,unsigned size) | 把p所指内存区的大小改为size个字节 | 新分配内存空间的地址；如不成功，返回0 |
| int rand(void) | 产生0～32767的随机整数 | 返回一个随机整数 |
| void exit(int state) | 程序终止执行，返回调用过程，state为0正常终止，非0非正常终止 | 无 |