

C 语言编程专题研究

2018 年版

郝立柱 郝立丽

应用统计教育部重点实验室
东北师范大学数学与统计学院

前 言

C 语言是在国内外广泛使用的一种计算机高级语言,它功能丰富、使用灵活便捷、表达能力强、应用面非常广泛、可移植性好,既有高级语言的特点,又具备低级语言的直接控制特性,非常适合于科学研究工作.

应用统计教育部重点实验室为了能够适应海量数据处理,要求所有的工作人员包括在籍学生必须掌握 C 语言,希望以 C 语言为根基,扩展到 C++ 从而适应科研的需要.

本书从编程的角度出发,共计十三个专题讨论,详细探究了基本编程技巧、基本编程技能及其基本编程方法.

全文使用 A4 纸张按照双栏排版,每一个专题主体占三页,作业占一页,学习或讲解每一个专题大约需要 90 分钟.学习时可以采用打印后,慢慢仔细研究编程规范和基本编程习惯.本书提供全部的程序文件以及相关辅助文件,这便于调试,理解程序运行过程中的每一步执行状态.

应用统计教育部重点实验室程序库主要使用 C/C++ 语言编写,这本书也是必要的入门提纲.

关于作者情况简述如下:郝立柱和郝立丽分别工作于东北师范大学数学与统计学院和东北林业大学经济管理学院,都是应用统计教育部重点实验室机器学习课题组成员,郝立柱的电子邮箱:HSsystem@163.com,如果读者有任何建议和意见欢迎来函批评指正.

郝立柱 郝立丽

2018 年 09 月 22 日

目 录

第 1 章 顺序结构.....	1
1.1 变量类型.....	1
1.2 表达式.....	1
1.3 输入.....	1
1.4 输出.....	1
1.5 简单计算.....	1
1.6 计算繁分式.....	1
1.7 函数计算.....	2
1.8 求方程的根.....	2
1.9 求二元一次方程组的解.....	2
1.10 计算三角形面积.....	3
1.11 整数除法余数问题.....	3
1.12 交换 a,b 之值.....	3
1.13 指数的计算技巧.....	3
1.14 作业.....	4
第 2 章 选择结构.....	5
2.1 判断语句格式.....	5
2.2 找出正整数 a,b 之中的最大者.....	5
2.3 找出正整数 a,b,c 三者中的最大者.....	5
2.4 将 a,b,c 按从小到大排序.....	6
2.5 判断谁大.....	6
2.6 整数的数位个数.....	6
2.7 switch 格式.....	7
2.8 各个月份的天数.....	7
2.9 作业.....	8
第 3 章 循环结构.....	9
3.1 while 格式.....	9
3.2 for 格式.....	9
3.3 3X+1 问题.....	9
3.4 求正整数 (a,b) 的最大公约数.....	9
3.5 二分法求方程的根.....	9
3.6 $S = 1 + 2 + 3 + \dots + 100$	9
3.7 $S = 1 * 2 + 2 * 3 + \dots + 99 * 100$	10
3.8 $S = \frac{1}{1} + \frac{1}{2} + \frac{1}{3} \dots + \frac{1}{100}$	10
3.9 $S = \frac{1}{1*2} + \frac{1}{2*3} + \frac{1}{3*4} \dots + \frac{1}{100*101}$	10

3.10	$S = \frac{1}{1} - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} \dots + \frac{1}{99} - \frac{1}{100}$	11
3.11	给一个十进制整数 a 求其位数	11
3.12	素数 (质数) 判断	11
3.13	作业	12
第 4 章	二重循环	13
4.1	图形打印	13
4.1.1	实心矩形	13
4.1.2	左三角实心图形	13
4.1.3	右三角实心图形	13
4.1.4	上三角实心图形	13
4.1.5	字母实心图形	14
4.1.6	数字实心图形	14
4.1.7	连续数字实心图形	14
4.2	计算三阶水仙花数	14
4.3	暴力搜索: 百钱买百鸡	15
4.4	级数求和	15
4.5	乘法口诀表	15
4.6	作业	16
第 5 章	一维数组	17
5.1	数组的定义	17
5.2	数组的初始化	17
5.3	替换数组中的元素	17
5.4	去掉数组中的某些元素	17
5.5	数组元素查询	17
5.6	数组元素插入	17
5.7	生成菲波纳切数列	18
5.8	分离十进制整数 a 的每一位数字	18
5.9	成绩均值与方差	18
5.10	运用数组打印空心图形	19
5.11	冒泡法排序	19
5.12	选择排序	19
5.13	集合的交集	19
5.14	作业	20
第 6 章	函数调用	21
6.1	函数定义	21
6.2	最小素因子函数	21
6.3	正整数的质因数分解	21
6.4	最大值函数	21

6.5 最大公因子函数.....	21
6.6 随机数发生器函数.....	22
6.7 组合函数.....	22
6.8 计算今天星期几.....	22
6.9 数组有序插入新值.....	23
6.10 作业.....	24
第 7 章 二维数组.....	25
7.1 数组的定义.....	25
7.2 数组的初始化.....	25
7.3 矩阵加法.....	25
7.4 矩阵的消法变换.....	25
7.5 学生成绩.....	25
7.6 获取子矩阵.....	26
7.7 化向量组为单位向量组.....	26
7.8 打印图形.....	27
7.9 奇数阶幻方.....	27
7.10 作业.....	28
第 8 章 堆栈技术.....	29
8.1 堆栈的一般形式.....	29
8.2 十进制转换为二进制.....	29
8.3 回文数.....	29
8.4 质因数分解.....	29
8.5 字符串反序.....	30
8.6 双侧堆栈.....	30
8.7 复杂连分数计算.....	30
8.8 数组数据翻转.....	31
8.9 利用堆栈走迷宫.....	31
8.10 作业.....	32
第 9 章 函数递归.....	33
9.1 十进制转换为二进制.....	33
9.2 菲波纳切数列的递归计算.....	33
9.3 递归计算杨辉三角.....	33
9.4 伞型序递归.....	33
9.5 利用递归求最大公因子.....	34
9.6 对策局势的确定.....	34
9.7 利用递归实现连分数计算.....	34
9.8 递归求根.....	34
9.9 递归实现全排列.....	35

9.10	递归搜索	35
9.11	作业	36
第 10 章	文件操作	37
10.1	文件操作的定义	37
10.2	写文件	37
10.3	数组内容写文件	37
10.4	读文件并存于数组中	37
10.5	将二维数组内容写入文件中	38
10.6	从文件中读入数据到二维数组中	38
10.7	不定长数据的存储	38
10.8	不定长数据的读取	39
10.9	文件合并	39
10.10	作业	40
第 11 章	指针初步	41
11.1	指针的定义	41
11.2	利用指针访问数组	41
11.3	数组元素倒序	41
11.4	交换两个元素	41
11.5	通用排序函数	42
11.6	动态内存开辟	42
11.7	不定长数据的读写	42
11.8	利用指针给二维数组赋值	43
11.9	C++ 上的动态空间开辟	43
11.10	指针与二维数组的关系	43
11.11	作业	44
第 12 章	再谈指针	45
12.1	定义指针数组	45
12.2	动态开辟二维数组	45
12.3	获取外部定长数据到内存	45
12.4	获取外部不定长数据到内存	46
12.5	指针数组的传递	46
12.6	指向指针的指针的指针	47
12.7	作业	48
第 13 章	结构体	49
13.1	结构体的定义	49
13.2	结构体的赋值	49
13.3	结构体的指针	49
13.4	简单链表	50
13.5	链表应用	50
13.6	C++ 的类定义	51
13.7	作业	52

第一章 顺序结构

§1.1 变量类型

```
int    i,j,k;    // 整型
long   m,n;      // 长整型
float  r,s;      // 浮点型实数
double t,x,y;    // 双精度实数
char   ch,c1,c2; // 字符型
```

§1.2 表达式

由四则运算符加减乘除以及括号和函数构成的式子, 就叫做表达式.

例如:

```
12*13+50
12+(34+23)*14-82/2
4*sin(1.013)+3*cos(2.023)
```

§1.3 输入

程序运行时系统会暂时停下来, 从键盘上获得用户提供的数据, 然后再继续运行.

例如: `scanf("%d",&g);`

运行时, 系统出现一个黑色的窗口, 然后屏幕上有一个闪动的光标, 此时, 用户需输入一个数, 比如:3 然后按回车. 系统获得数据后, 自动将这个数存入变量 `g` 中.

如果输入两个数据, 分别存入变量 `g` 和 `h` 中, 则程序如下:

```
scanf("%d%d",&g,&h);
```

运行时, 输入两个数, 中间用空格隔开, 比如:3 4

§1.4 输出

程序运行时从系统输出到屏幕上的信息.

```
printf("%d",g);
```

如果想输出两个变量的值, 并且两个值之间用逗号分隔, 如:

```
printf("%d,%d",g,h);
```

输出的格式有很多, 要求掌握.

```
%d    // 整型 int 定义的变量
%ld    // 长整型 long 定义的变量
%f    // 浮点型实数 float 定义的变量
%lf    // 双精度实数 double 定义的变量
%e    // 指数形式实数 float 定义的变量
%le    // 指数形式实数 double 定义的变量
%c    // 字符型 char 定义的变量
%s    // 字符串 char * 定义的变量
%s    // 字符数组
\n    // 回车换行符
```

§1.5 简单计算

计算: $124*126$

```
程序名称:C01_010.c
#include "stdio.h"
#include "stdlib.h"
```

```
void main() {
    printf("%d",124*126);
}
```

结果为:15624

从键盘输入两个数, 计算并输出其乘积.

```
程序名称:C01_020.c
#include "stdio.h"
#include "stdlib.h"
```

```
void main() {
    double a,b;

    scanf("%lf%lf",&a,&b);
    printf("%lf\n",a*b);
}
```

输入:4 5

输出:20.000000

§1.6 计算繁分式

$$\frac{11}{12 + \frac{3}{23}}$$

```
程序名称:C01_030.c
#include "stdio.h"
#include "stdlib.h"
```

```
void main() {
    printf("%lf", 11.0/(12.0+3.0/23.0));
}
```

结果为:0.906810

也可以, 这样写程序:

```
程序名称:C01_040.c
#include "stdio.h"
#include "stdlib.h"
```

```
void main() {
    double x,y;

    x=3.0/23.0;
    y=11.0/(12.0+x);
    printf("%lf",y);
}
```

但是, 不能这样写程序:

```
printf("%lf", 11/(12+3/23));
```

结果为:0.000000 或者显示错误

由于整数除以整数, 结果是整数, 小数部分会自动删除, 注意系统不会四舍五入, 这是初学者常犯的错误之一。

如果您想计算下面的结果:

$$\left[\frac{110}{12 + \left[\frac{30}{23} \right]} \right]$$

这里 $[\cdot]$ 代表高斯函数, 即数论上的取整函数。

```
程序名称:C01_050.c
#include "stdio.h"
#include "stdlib.h"
```

```
void main() {
    printf("%d", 110/(12+30/23));
}
```

结果为:8

注意, 本题输出时, 需要使用 `%d` 而不能使用 `%f`, 因为输出的是整数, 所以必须用 `%d` 或者 `%ld`, 否则可能显示莫名其妙的杂乱文字。

§1.7 函数计算

求值 $\sin(28.5^\circ) + \cos(12.3^\circ) + \sqrt{2.4}$

程序名称:C01_060.c

```
#include "stdio.h"
#include "stdlib.h"
#include "math.h"

void main() {
    double x,y;

    x=sin(28.5/180*3.1415926);
    y=cos(12.3/180*3.1415926);
    printf("%lf", x+ y + sqrt(2.4));
}
```

结果为:3.003398

§1.8 求方程的根

已知方程有两个不相同的实根, 求方程

$$x^2 - 5x + 3 = 0$$

的根。

```
程序名称:C01_070.c
#include "stdio.h"
#include "stdlib.h"
#include "math.h"

void main() {
    double a,b,c,x1,x2;
    a=1; b=-5; c=3;
    x1=(-b+sqrt(b*b-4*a*c))/(2*a);
    x2=(-b-sqrt(b*b-4*a*c))/(2*a);
    printf("x1=%lf\n", x1);
    printf("x2=%lf\n", x2);
}
```

结果为:

x1=4.302776
x2=0.697224

§1.9 求二元一次方程组的解

计算二元一次方程组的解:

$$\begin{cases} 3x_1 + 4x_2 = 5 \\ 2x_1 + x_2 = 3 \end{cases}$$

其解为:

$$x_1 = \frac{\begin{vmatrix} c_1 & a_{12} \\ c_2 & a_{22} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}}, \quad x_2 = \frac{\begin{vmatrix} a_{11} & c_1 \\ a_{21} & c_2 \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}}$$

程序名称:C01_080.c

```
#include "stdio.h"
```

```
#include "stdlib.h"
```

```
void main() {
    double a11,a12,a21,a22,c1,c2,x1,x2;
    a11=3; a12=4; c1=5;
    a21=2; a22=1; c2=3;
    x1=(c1*a22-c2*a12)/(a11*a22-a12*a21);
    x2=(a11*c2-a21*c1)/(a11*a22-a12*a21);
    printf("x1=%lf\n", x1);
    printf("x2=%lf\n", x2);
}
```

结果为:

x1=1.400000

x2=0.200000

§1.10 计算三角形面积

已知三角形的两边及其夹角, 求三角形面积, 其中 $a = 3, b = 4, \angle C = 27.3^\circ$. 面积

$$S = \frac{1}{2}ab \sin C$$

程序名称:C01_090.c

```
#include "stdio.h"
```

```
#include "stdlib.h"
```

```
#include "math.h"
```

```
void main() {
    double a,b,C,s;
    a=3; b=4;
    C=27.5/180*3.1415926;
    s=0.5*a*b*sin(C);
    printf("s=%lf\n", s);
}
```

结果为:s=2.770492

§1.11 整数除法余数问题

问 1234 除以 19 余数是多少?

程序名称:C01_100.c

```
#include "stdio.h"
```

```
#include "stdlib.h"
```

```
void main() {
    int x;
    x=1234%19;
    printf("%d\n",x);
}
```

结果为:18

§1.12 交换 a,b 之值

原来是 $a = 3, b = 4$ 交换 a,b 之内容变为 $a = 3, b = 4$.

显然不能直接 $a = b$, 否则在将 b 的内容放入 a 中时, 系统会清除 a 中的一切内容.

正确的做法是用中间变量来缓存一下. 如 $t = a, a = b, b = t$

程序名称:C01_110.c

```
#include "stdio.h"
```

```
#include "stdlib.h"
```

```
void main() {
    int a,b,t;
    scanf("%d,%d",&a,&b);
    printf("a=%d,b=%d\n",a,b);
    t=a; a=b; b=t;
    printf("a=%d,b=%d\n",a,b);
}
```

输入:3,4

结果为:

a=3,b=4

a=4,b=3

§1.13 指数的计算技巧

计算 $\ln(2^{14} + 3^{24})$.

数值较小, 可以直接计算, 系统仅提供了 e 的指数函数, 所以需要简单的变换:

$\log(\exp(14*\log(2))+\exp(24*\log(3)))$

计算 $\ln(2^{1234} + 3^{4567})$.

本题不能直接计算, 否则中间步骤会出现双精度数超界. 令 $2^{1234} = e^a, 3^{4567} = e^b$, 计算出 a, b 的值. 这样原题转化为 $\ln(e^a + e^b)$, 不难看出 $a < b$, 于是 $\ln(e^b(e^{a-b} + 1))$, 即得 $b + \ln(e^{a-b} + 1)$

程序名称:C01_120.c

```
#include "stdio.h"
```

```
#include "stdlib.h"
```

```
#include "math.h"
```

```
void main() {
    double a,b,s;
    a=1234*log(2);
    b=4567*log(3);
    s=b+log(exp(a-b)+1);
    printf("s=%lf\n",s);
}
```

结果为:s=5017.362322

§1.14 作业

1. 函数计算

$$e^{\sin(18^\circ)} \log_{10} 42 + e^{12 \ln(13)}$$

2. 计算繁分式

$$\frac{11}{12 + \frac{333}{13 + \frac{23}{5}}}$$

3. 求三元一次方程组的解

$$\begin{cases} x_1 + x_2 + x_3 = 1 \\ 2x_1 + 2x_2 + x_3 = 3 \\ 3x_1 + 4x_2 + 5x_3 = 5 \end{cases}$$

4. 计算三角形面积

已知三角形的三边的长度, 求三角形面积,
其中 $a = 3, b = 4, c = 6$. 面积公式

$$S = \sqrt{p(p-a)(p-b)(p-c)}$$

其中 p 为半周长, 即 $p = \frac{1}{2}(a+b+c)$.

5. 整数除法余数问题, 问:

1234 除以 11 的余数乘以

1234 除以 17 的余数乘以

1234 除以 19 的余数等于几?

6. 现有一张考卷, 共有 50 道选择题, 规定选对一题得 2 分, 选错 (或不选) 一题扣 1 分。假设考卷的成绩大于零分而不超过十分, 请你估计能有多少题答对了。(手动推理, 程序直接给出结果即可)

7*. 求 $\ln(\ln(2^{3^{6872}+7^{4116}+5^{2237}} + 7^{3^{2764}+5^{2217}}))$
本题较难, 选作.

第二章 选择结构

§2.1 判断语句格式

第一种形式:

```
if (条件) {  
    程序块1  
}
```

第二种形式:

```
if (条件) {  
    程序块1  
} else {  
    程序块2  
}
```

第三种形式:

```
if (条件) {  
    程序块1  
} else if (条件) {  
    程序块2  
} else if (条件) {  
    程序块3  
} else {  
    程序块4  
}
```

§2.2 找出正整数 a,b 之中的最大者

比如:

输入 3,6 输出 6
输入 5,4 输出 5

程序名称:C02_010.c
#include "stdio.h"
#include "stdlib.h"

```
void main() {  
    int a,b;  
  
    scanf("%d,%d",&a,&b);  
    if (a<=b) {  
        printf("max=%d\n", b);  
    } else {  
        printf("max=%d\n", a);  
    }  
}  
输入:3,6  
输出:max=6
```

§2.3 找出正整数 a,b,c 三者中的最大者

比如:

输入 3,6,2 输出 6
输入 5,4,8 输出 8

程序名称:C02_020.c
#include "stdio.h"
#include "stdlib.h"

```
void main() {  
    int a,b,c;  
  
    scanf("%d,%d,%d",&a,&b,&c);  
    if (a<=b) {  
        if (b<=c) {  
            printf("max=%d\n", c);  
        } else {  
            printf("max=%d\n", b);  
        }  
    } else {  
        if (a<=c) {  
            printf("max=%d\n", c);  
        } else {  
            printf("max=%d\n", a);  
        }  
    }  
}  
输入:3,6,2  
输出:max=6
```

还可以采用暂存中间结果的方式实现.

程序名称:C02_030.c
#include "stdio.h"
#include "stdlib.h"

```
void main() {  
    int a,b,c,d;  
  
    scanf("%d,%d,%d",&a,&b,&c);  
    if (a<=b) {  
        if (b<=c) d=c; else d=b;  
    } else {  
        if (a<=c) d=c; else d=a;  
    }  
    printf("max=%d\n", d);  
}
```

还有更加简单的形式,使用 C 语言提供的三目运算符来实现.

程序名称:C02_040.c
#include "stdio.h"
#include "stdlib.h"

```

void main() {
    int a,b,c,d;

    scanf("%d,%d,%d",&a,&b,&c);
    if (a<=b) {
        d= b<=c ? c:b;
    } else {
        d= a<=c ? c:a;
    }
    printf("max=%d\n", d);
}

```

§2.4 将 a,b,c 按从小到大排序

比如:

输入 3,6,2 输出 2,3,6

输入 5,4,1 输出 1,4,5

程序名称:C02_050.c

```

#include "stdio.h"
#include "stdlib.h"
#include "math.h"

```

```

void main() {
    int a,b,c,t;

    scanf("%d,%d,%d",&a,&b,&c);
    if (a>b) {
        t=a; a=b; b=t;
    }
    if (b>c) {
        t=b; b=c; c=t;
    }
    if (a>b) {
        t=a; a=b; b=t;
    }
    printf("%d,%d,%d\n",a,b,c);
}

```

输入:3,6,2

输出:2,3,6

§2.5 判断谁大

判断 $\sin(x)$ 与 $x - (1/6)x^3$ 谁大?

其中 $x = 1.2345$

程序名称:C02_060.c

```

#include "stdio.h"
#include "stdlib.h"
#include "math.h"

```

```

void main() {
    double x;

    x=1.2345;
    if (sin(x)>x-(1.0/6)*x*x*x) {
        printf("sin(x) 大\n");
    } else {
        printf("sin(x) 小\n");
    }
}

```

结果为:sin(x) 大

判断 $1+x$ 与 $e^{\frac{x}{1+x}}$ 谁大?

其中 $x = 1.12345$

程序名称:C02_070.c

```

#include "stdio.h"
#include "stdlib.h"
#include "math.h"

```

```

void main() {
    double x;

    x=1.12345;
    if (1+x>exp(x/(1+x))) {
        printf("1+x 大\n");
    } else {
        printf("1+x 小\n");
    }
}

```

结果为:1+x 大

§2.6 整数的数位个数

问正整数 x 是由十进制几位数字组成?

其中 $1 \leq x \leq 99999$

比如:

输入 123 输出 3

输入 2345 输出 4

程序名称:C02_080.c

```

#include "stdio.h"
#include "stdlib.h"

```

```

void main() {
    long x;

    scanf("%d",&x);
    if (x<=9) {
        printf("1");
    } else if (x<=99) {
        printf("2");
    } else if (x<=999) {

```

```

    printf("3");
} else if (x<=9999) {
    printf("4");
} else {
    printf("5");
}
}

```

输入:3344

输出:4

§2.7 switch 格式

```

switch(表达式) {
    case 常量表达式1:
        程序块1
        break;
    case 常量表达式2:
        程序块2
        break;
    ... ..
    default:
        程序块n
}

```

§2.8 各个月份的天数

键盘输入月份,系统输出该月份所含有的天数.(不考虑闰年)

使用 if 的方式实现.

程序名称:C02_090.c

```
#include "stdio.h"
```

```
#include "stdlib.h"
```

```

int main() {
    int x,d;

    scanf("%d",&x);
    if (x==4 || x==6 || x==9 || x==11) {
        d=30;
    } else if (x==2) {
        d=28;
    } else {
        d=31;
    }
    printf("%d\n",d);
}

```

使用 switch 的方式实现.

程序名称:C02_100.c

```
#include "stdio.h"
```

```

#include "stdlib.h"

int main() {
    int x,d;
    scanf("%d",&x);
    switch (x) {
        case 1: d=31; break;
        case 2: d=28; break;
        case 3: d=31; break;
        case 4: d=30; break;
        case 5: d=31; break;
        case 6: d=30; break;
        case 7: d=31; break;
        case 8: d=31; break;
        case 9: d=30; break;
        case 10: d=31; break;
        case 11: d=30; break;
        case 12: d=31; break;
    }
    printf("%d\n",d);
}

```

还可以利用 switch 的特性简化程序.

程序名称:C02_110.c

```
#include "stdio.h"
```

```
#include "stdlib.h"
```

```

int main() {
    int x,d;
    scanf("%d",&x);
    switch (x) {
        case 4:
        case 6:
        case 9:
            d=30; break;
        case 2: d=28; break;
        default: d=31;
    }
    printf("%d\n",d);
}

```

另外可以利用三目运算符来实现.

程序名称:C02_120.c

```
#include "stdio.h"
```

```
#include "stdlib.h"
```

```

int main() {
    int x,d;
    scanf("%d",&x);
    d=x==4||x==6||x==9||x==11?30:(x==2?28:31);
    printf("%d\n",d);
}

```

输入:2

输出:28

§2.9 作业

1. 找出正整数 a,b,c,d 四者中的最大者

2. 键盘输入四个正整数, 将这四个正整数 a,b,c,d 按从大到小排序.

3. 判断 $\tan(x)$ 与 $x - (1/3)x^3$ 谁大?

其中 $x = 1.2345$

4. 判断 $\frac{1}{1-x}$ 与 $e^{\frac{x}{1-x}}$ 谁大?

其中 $x = 0.12345$

5. 符号函数

$$\begin{cases} -1 & (x < 0) \\ 0 & (x = 0) \\ 1 & (x > 0) \end{cases}$$

键盘输入一个整数, 输出其符号, 即 -1, 0, 1.

6. 求方程的根

$$ax^2 + bx + c = 0$$

其中 a,b,c 为实数, 并由键盘输入.

无实数根时, 输出“无解”;

有单一实根时, 输出“ $x_1 = x_2 =$ ”并输出那个解;

若有两个不同的实根时, 输出每一个实根.

7. 键盘输入年份和月份, 例如:199801, 表示 1998 年 1 月, 系统输出该月份所含有的天数. 下面的表达式 d 为 1 则表示 x 为闰年, d 为 0 则表示为非闰年.

$$d = x\%400==0 \text{ || } (x\%100!=0 \text{ \&\& } x\%4==0);$$

8. 每吨货物每公里费用 P 与距离 S 有关, 路途越远, 每公里运价越低, 其运费与距离关系如下:

$$p = \begin{cases} 10 & s < 100 \\ 8 & 100 \leq s < 150 \\ 7 & 150 \leq s < 200 \\ 6 & 200 \leq s < 300 \\ 5.5 & s \geq 300 \end{cases}$$

从键盘上输入货物吨数, 运输公里数, 求应付的运费。

第三章 循环结构

§3.1 while 格式

```
while (条件) {  
    循环体  
}
```

§3.2 for 格式

```
for (初值; 条件; 增量) {  
    循环体  
}
```

§3.3 $3X+1$ 问题

从键盘输入一个正整数 x , 然后按照下面的规则, 进行迭代, 直至出现 1 为止.

$$f(x) = \begin{cases} 3x+1 & \text{当 } x \text{ 为奇数时} \\ x/2 & \text{当 } x \text{ 为偶数时} \end{cases}$$

例如: 输入 6, 系统输出

6,3,10,5,16,8,4,2,1

程序名称:C03_010.c

```
#include "stdio.h"
```

```
#include "stdlib.h"
```

```
void main() {  
    int x;  
  
    scanf("%d",&x);  
    while (x!=1) {  
        printf("%d,",x);  
        if (x%2==0) x/=2; else x=3*x+1;  
    }  
    printf("%d\n",x);  
}
```

输入:17

输出:17,52,26,13,40,20,10,5,16,8,4,2,1

§3.4 求正整数 (a,b) 的最大公约数

使用辗转相除法求最大公约数, 即若 $a > b$, 则 a 除以 b 后的余数存入 a 中, 否则 b 除以 a 后的余数存入 b 中, 反复迭代直至 a, b 中有一个为零时停止.

程序名称:C03_020.c

```
#include "stdio.h"
```

```
#include "stdlib.h"
```

```
void main() {  
    int a,b;  
  
    scanf("%d,%d",&a,&b);  
    while (a>0 && b>0) {  
        if (a>b) a%=b; else b%=a;  
    }  
    printf("%d\n",a+b);  
}
```

输入:56,63

输出:7

§3.5 二分法求方程的根

已知函数

$$f(x) = x^3 - 5x + 1, \quad f(0) > 0, f(1) < 0$$

由于函数 $f(x)$ 连续, 可见在 $(0,1)$ 区间上必然存在一个根, 求其根. 精确到小数点后 3 位, 即 0.001.

程序名称:C03_030.c

```
#include "stdio.h"
```

```
#include "stdlib.h"
```

```
void main() {  
    double a,b,c;  
  
    a=0;  
    b=1;  
    while (b-a>0.001) {  
        c=(a+b)/2.0;  
        if (c*c*c-5*c+1>0) a=c; else b=c;  
    }  
    printf("%.3lf\n",a);  
}
```

结果为:0.201

§3.6 $S = 1 + 2 + 3 + \dots + 100$

1. 产生 1,2,3,4,..., 方法是 $i = i + 1$;
 2. 累加 1,1+2,1+2+3,..., 方法是 $s = s + i$;
- 使用 while 循环实现

程序名称:C03_040.c
#include "stdio.h"
#include "stdlib.h"

```
void main() {
    int i,s;

    s=0;
    i=1;
    while (i<=100) {
        s=s+i;
        i=i+1;
    }
    printf("%d\n",s);
}
```

结果为:5050

使用 for 循环实现

程序名称:C03_050.c
#include "stdio.h"
#include "stdlib.h"

```
void main() {
    int i,s;

    s=0;
    for(i=1; i<=100; i=i+1) {
        s=s+i;
    }
    printf("%d\n",s);
}
```

§3.7 $S = 1 * 2 + 2 * 3 + \dots + 99 * 100$

1. 产生 1,2,3,4,..., 方法是 $i = i + 1$;
2. 计算中间结果 $1 * 2, 2 * 3, \dots$, 方法是 $t = i * (i + 1)$;
3. 累加 $s = 1 * 2, s = s + 2 * 3, s = s + 3 * 4, \dots$, 方法是 $s = s + t$;

使用 while 循环实现

程序名称:C03_060.c
#include "stdio.h"
#include "stdlib.h"

```
void main() {
    int i,s,t;

    s=0;
    i=1;
    while (i<100) {
```

```
        t=i*(i+1);
        s=s+t;
        i=i+1;
    }
    printf("%d\n",s);
}
```

结果为:333300

使用 for 循环实现

程序名称:C03_070.c
#include "stdio.h"
#include "stdlib.h"

```
void main() {
    int i,s,t;

    s=0;
    for(i=1; i<100; i=i+1) {
        t=i*(i+1);
        s=s+t;
    }
    printf("%d\n",s);
}
```

§3.8 $S = \frac{1}{1} + \frac{1}{2} + \frac{1}{3} \dots + \frac{1}{100}$

程序名称:C03_080.c
#include "stdio.h"
#include "stdlib.h"

```
void main() {
    int i;
    double s,t;

    s=0.0;
    for(i=1; i<=100; i=i+1) {
        t=1.0/i;
        s=s+t;
    }
    printf("%lf\n",s);
}
```

结果为:5.187378

§3.9 $S = \frac{1}{1*2} + \frac{1}{2*3} + \frac{1}{3*4} \dots + \frac{1}{100*101}$

程序名称:C03_090.c
#include "stdio.h"
#include "stdlib.h"

```
void main() {
    int i;
    double s,t;
```



```

s=0.0;
for(i=1; i<=100; i=i+1) {
    t=1.0/(i*(i+1));
    s=s+t;
}
printf("%lf\n",s);
}

```

结果为:0.990099

$$\S 3.10 \quad S = \frac{1}{1} - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} \dots + \frac{1}{99} - \frac{1}{100}$$

```

程序名称:C03_100.c
#include "stdio.h"
#include "stdlib.h"

void main() {
    int i,k;
    double s,t;

    s=0.0;
    k=1;
    for(i=1; i<=100; i=i+1) {
        t=(double)k/i;
        s=s+t;
        k=-k;
    }
    printf("%lf\n",s);
}

```

结果为:0.688172

还有一种处理交替符号的方法,这种方法还可以进一步推广为处理任意具有周期特征的符号现象.

```

程序名称:C03_110.c
#include "stdio.h"
#include "stdlib.h"

void main() {
    int i,k;
    double s,t;

    s=0.0;
    k=0;
    for(i=1; i<=100; i=i+1) {
        if (k==0) {
            t=1.0/i;
        } else {
            t=-1.0/i;
        }
        s=s+t;
        k=k+1; if (k>=2) k=k-2;
    }
}

```

```

}
printf("%lf\n",s);
}

```

§3.11 给一个十进制整数 a 求其位数

比如:

输入 12345 系统输出 5

输入 3456789 系统输出 7

程序名称:C03_120.c

```

#include "stdio.h"
#include "stdlib.h"

```

```

void main() {
    int i,x;

    scanf("%d",&x);
    for (i=0; x>0; i++) x=x/10;
    printf("%d\n",i);
}

```

输入:3456789
输出:7

§3.12 素数(质数)判断

从键盘输入一个大于 1 的正整数,编程判断其是否为素数.

比如:

输入 12 系统输出 No

输入 91 系统输出 No

输入 101 系统输出 Yes

程序名称:C03_130.c

```

#include "stdio.h"
#include "stdlib.h"
#include "math.h"

```

```

void main() {
    int i,x;

    scanf("%d",&x);
    for (i=2; i<x; i++) {
        if (x%i==0) break;
    }
    if (i>=x) printf("Yes"); else printf("No");
}

```

输入:91
输出:No

§3.13 作业

1. 伪随机数的产生

初始值 $x = 1$, 按照下面的规则进行迭代:

$$f(x) = (x * 314159269 + 453806245) \% 2147483648$$

计算出 $f(x)$ 后, 将 $f(x)$ 再赋值给 x , 然后输出

$$r = x / 2147483648.0$$

迭代 10 次, 输出前 10 项的各个 r 值.

2. 求正整数 (a,b,c) 的最大公约数.

3. 二分法求方程的根

已知函数

$$f(x) = \sin(x) + 2x - 1, \quad f(0) < 0, f(1) > 0$$

由于函数 $f(x)$ 连续, 可见在 $(0,1)$ 区间上必然存在一个根, 求其根. 精确到小数点后 2 位, 即 0.01.

4. 计算数列和.

$$S = \frac{1}{1} + \frac{1}{2} - \frac{1}{3} - \frac{1}{4} \cdots - \frac{1}{99} - \frac{1}{100}$$

符号为 $++--++--\cdots$, 先两个正数, 然后两个负数, 循环往复.

5. 余数问题 (中国剩余定理)

今有物不知其数, 三三数之剩二, 五五数之剩三, 七七数之剩二, 问物几何?

(提示: 从 1 到 ∞ 逐一测试, 直到找到满足上述条件的数为止)

第四章 二重循环

§4.1 图形打印

§4.1.1 实心矩形

打印如下图形:

```
*****
*****
*****
*****
*****
```

程序名称:C04_010.c

```
#include "stdio.h"
#include "stdlib.h"
```

```
void main() {
    int i;
    for (i=0; i<5; i++) {
        printf("*****\n");
    }
}
```

以上是一次打印一行,这种方式不够灵活,可以再分细,每一行中每次仅打印一个字符,也就是说再用一重循环,可以控制到每次打印一个字符的目的.

程序名称:C04_020.c

```
#include "stdio.h"
#include "stdlib.h"
```

```
void main() {
    int i,j;
    for (i=0; i<5; i++) {
        for (j=0; j<10; j++) {
            printf("*");
        }
        printf("\n");
    }
}
```

§4.1.2 左三角实心图形

打印如下图形:

```
*
**
***
****
*****
```

程序名称:C04_030.c

```
#include "stdio.h"
#include "stdlib.h"
```

```
void main() {
    int i,j;

    for (i=0; i<5; i++) {
        for (j=0; j<i+1; j++) printf("*");
        printf("\n");
    }
}
```

§4.1.3 右三角实心图形

打印如下图形:

```
*
**
***
****
*****
```

本题在打印时,由于很难控制光标在屏幕上的位置,所以最简单的方法是通过打印一些空白字符来占位,从而实现对齐图形的目的.

程序名称:C04_040.c

```
#include "stdio.h"
#include "stdlib.h"
```

```
void main() {
    int i,j;

    for (i=0; i<5; i++) {
        for (j=0; j<5-i; j++) printf(" ");
        for (j=0; j<i+1; j++) printf("*");
        printf("\n");
    }
}
```

§4.1.4 上三角实心图形

打印如下图形:

```
*
**
***
****
*****
*****
*****
```

程序名称:C04_050.c

```
#include "stdio.h"
#include "stdlib.h"
```

```
void main() {
    int i,j;
```

```

for (i=0; i<5; i++) {
    for (j=0; j<5-i; j++) printf(" ");
    for (j=0; j<2*i+1; j++) printf("*");
    printf("\n");
}
}

```

§4.1.5 字母实心图形

打印如下图形:

```

a
bbb
cccc
ddddd
eeeeeee

```

程序名称:C04_060.c
#include "stdio.h"
#include "stdlib.h"

```

void main() {
    int i,j;
    char ch;
    ch='a';
    for (i=0; i<5; i++,ch++) {
        for (j=0; j<5-i; j++) printf(" ");
        for (j=0; j<2*i+1; j++) printf("%c",ch);
        printf("\n");
    }
}

```

§4.1.6 数字实心图形

打印如下图形:

```

1
123
12345
1234567
123456789

```

程序名称:C04_070.c
#include "stdio.h"
#include "stdlib.h"

```

void main() {
    int i,j;
    char ch;
    for (i=0; i<5; i++) {
        for (j=0; j<5-i; j++) printf(" ");
        ch='1';
        for (j=0; j<2*i+1; j++,ch++) {
            printf("%c",ch);
        }
        printf("\n");
    }
}

```

§4.1.7 连续数字实心图形

打印如下图形:

```

      1
    2 3 4
  5 6 7 8 9
10 11 12 13 14 15 16
17 18 19 20 21 22 23 24 25

```

程序名称:C04_080.c

#include "stdio.h"

#include "stdlib.h"

```

void main() {
    int i,j,k;
    k=1;
    for (i=0; i<5; i++) {
        for (j=0; j<5-i; j++) printf(" ");
        for (j=0; j<2*i+1; j++,k++) {
            printf("%4d",k);
        }
        printf("\n");
    }
}

```

§4.2 计算三阶水仙花数

一个三位数,若其各个数位上的数的立方和恰好等于该数本身,就称之为三阶水仙花数。例如:

$$153 = 1^3 + 5^3 + 3^3$$

求所有的三阶水仙花数。

程序名称:C04_090.c

#include "stdio.h"

#include "stdlib.h"

```

void main() {
    int i,j,k,m;
    for (i=1; i<10; i++) {
        for (j=0; j<10; j++) {
            for (k=0; k<10; k++) {
                m=i*i*i+j*j*j+k*k*k;
                if (i*100+j*10+k==m) {
                    printf("%d,",i*100+j*10+k);
                }
            }
        }
    }
}

```

结果为:153,370,371,407,

§4.3 暴力搜索: 百钱买百鸡

鸡翁一值钱五, 鸡母一值钱三, 鸡雏三值钱一, 百钱买百鸡, 问鸡翁母雏各几何?

显然, 鸡雏的个数一定是 3 的倍数, 故令鸡雏从 0 搜索到 99, 鸡翁从 0 搜索到 20, 鸡母只需 100-鸡翁-鸡雏即可。

程序名称:C04_100.c

```
#include "stdio.h"
```

```
#include "stdlib.h"
```

```
void main() {
    int i,j,k; // i:鸡翁,j:鸡母,k:鸡雏
    for (i=0; i<=20; i++) {
        for (k=0; k<=99; k+=3) {
            j=100-i-k;
            if (j>=0 && i*5+j*3+k/3==100) {
                printf("%d,%d,%d\n",i,j,k);
            }
        }
    }
}
```

结果为:

0,25,75

4,18,78

8,11,81

12,4,84

§4.4 级数求和

$$S = \frac{1}{1} + \frac{1}{1+2} + \frac{1}{1+2+3} + \cdots + \frac{1}{1+2+\cdots+100}$$

程序名称:C04_110.c

```
#include "stdio.h"
```

```
#include "stdlib.h"
```

```
void main() {
    int i,j;
    double s,t;

    s=0.0;
    for (i=1; i<=100; i++) {
        t=0.0;
        for (j=1; j<=i; j++) t+=j;
        s+=1.0/t;
    }
    printf("%lf\n",s);
}
```

结果为:1.980198

如果存储中间结果, 则程序的效率会更高。通常情况, 对于能够提高程序效率的中间结果, 一般都加以利用。

程序名称:C04_120.c

```
#include "stdio.h"
```

```
#include "stdlib.h"
```

```
void main() {
    int i;
    double s,t;

    s=0.0;
    t=0.0;
    for (i=1; i<=100; i++) {
        t+=i;
        s+=1.0/t;
    }
    printf("%lf\n",s);
}
```

§4.5 乘法口诀表

打印如下图形:

```
1*1=1
1*2=2   2*2=4
1*3=3   2*3=6   3*3=9
1*4=4   2*4=8   3*4=12  4*4=16
1*5=5   2*5=10  3*5=15  4*5=20  5*5=25
... ..
```

程序名称:C04_130.c

```
#include "stdio.h"
```

```
#include "stdlib.h"
```

```
void main() {
    int i,j;

    for (i=1; i<=9; i++) {
        for (j=1; j<=i; j++) {
            printf("%d*%d=%-4d",j,i,j*i);
        }
        printf("\n");
    }
}
```

§4.6 作业

1. 打印如下图形:

```
*****
****
***
**
*
```

2. 打印如下图形:

```
*****
****
***
**
*
```

3. 打印如下图形:

```
      *
     ***
    *****
   *****
  *****
 *****
 *****
  ***
   *
```

4. 打印如下图形:

```
aaaa*
aaa**
aa***
a****
```

5. 打印如下图形:

```
aaaa*aaaa
aaa***aaa
aa*****aa
a*****a
```

6. 打印如下图形:

```
      *      *
     ***    ***
    *****  *****
   *****  *****
  *****  *****
 *****  *****
*****  *****
*****  *****
*****  *****
***      ***
*        *
```

7. 打印如下图形:

```
      *      *
     ***    ***
    *****  *****
   *****  *****
  *****  *****
 *****  *****
*****  *****
*****  *****
*****  *****
***      ***
*        *
```

8. 打印如下图形:

```
      a
     abc
    abcde
   abcdefg
  abcdefghi
 abcdefg
  abcde
   abc
    a
```

9. 打印如下图形:

```
      a
     aba
    abcba
   abcdcba
  abcdedcba
 abcdcba
  abcba
   aba
    a
```

10. 计算四阶水仙花数

一个四位数, 若其各个数位上的数的四次方和恰好等于该数本身, 就称之为四阶水仙花数. 例如:

$$1634 = 1^4 + 6^4 + 3^4 + 4^4$$

求所有的四阶水仙花数.

11. 输出 100 以内的全部素数.(参见上节素数判断)

12. 暴力搜索满足方程 $3x + 5y = 98$ 的全部正整数解.

13. 级数求和: $S = 1! + 2! + 3! + \cdots + 7!$

14. 求值 $\frac{1}{1!} - \frac{1}{2!} + \frac{1}{3!} - \frac{1}{4!} + \cdots + \frac{1}{9!} - \frac{1}{10!}$

15. 求 1 角、2 角和 5 角钱币组成 1 元的所有组合.

第五章 一维数组

§5.1 数组的定义

```
int    ab[15];
double bc[30];
char   kk[10];
```

§5.2 数组的初始化

```
int    ab[15]={1,5,3,6,8,9,3,2,0,4};
double bc[30]={12.1,13.1,14.5};
char   kk[10]='a','b','m','1','2','0';
```

§5.3 替换数组中的元素

已知数组:

1, 3, 4, 7, 8, 10, 13, 14, 16, 17, 19, 21, 23, 29, 30

编写程序, 将上述数组中的偶数全部替换成 0, 要求不改变原数组元素的顺序.

```
程序名称:C05_010.c
#include "stdio.h"
#include "stdlib.h"
```

```
void main() {
    int i;
    int aa[15]={1,3,4,7,8,10,13,14,16,17,19,
                21,23,29,30};

    for (i=0; i<15; i++) {
        if (aa[i]%2==0) aa[i]=0;
    }
    for (i=0; i<15; i++) {
        printf("%d,",aa[i]);
    }
}
结果为:1,3,0,7,0,0,13,0,0,17,19,21,23,29,0,
```

§5.4 去掉数组中的某些元素

已知数组:

1, 3, 4, 7, 8, 10, 13, 14, 16, 17, 19, 21, 23, 29, 30

编写程序, 将上述数组中的偶数全部去掉, 得到新的数组:

1, 3, 7, 13, 17, 19, 21, 23, 29

要求不改变原数组元素的顺序.

程序名称:C05_020.c

```
#include "stdio.h"
#include "stdlib.h"
```

```
void main() {
    int i,j;
    int aa[15]={1,3,4,7,8,10,13,14,16,17,19,
                21,23,29,30};

    for (i=0,j=0; i<15; i++) {
        if (aa[i]%2!=0) { aa[j]=aa[i]; j++; }
    }
    for (i=0; i<j; i++) {
        printf("%d,",aa[i]);
    }
}
结果为:1,3,7,13,17,19,21,23,29,
```

§5.5 数组元素查询

已知数组具有从小到大的顺序关系:

1, 3, 4, 7, 8, 10, 13, 14, 16, 17, 19, 21, 23, 29, 30

键盘输入一个数 x, 判断其是否在数组中.

```
程序名称:C05_030.c
#include "stdio.h"
#include "stdlib.h"
```

```
void main() {
    int i,x;
    int aa[15]={1,3,4,7,8,10,13,14,16,17,19,
                21,23,29,30};

    scanf("%d",&x);
    for (i=0; i<15; i++) {
        if (x==aa[i]) break;
    }
    if (i<15) printf("Yes"); else printf("No");
}
输入:8
输出:Yes
```

由于本题已知数据的有序性, 故可以使用二分法查找.

§5.6 数组元素插入

已知数组具有从小到大的顺序关系:

1, 3, 4, 7, 8, 10, 13, 14, 16, 17, 19, 21, 23

键盘输入一个数 x , 将其插入到数组中, 并且保持原有顺序.

程序名称:C05_040.c

```
#include "stdio.h"
```

```
#include "stdlib.h"
```

```
void main() {
    int i,n,x;
    int aa[20]={1,3,4,7,8,10,13,14,16,17,19,
                21,23};

    n=12;
    scanf("%d",&x);
    for (i=n; i>=0; i--) {
        if (aa[i]>x) aa[i+1]=aa[i]; else break;
    }
    aa[i+1]=x;
    for (n++,i=0; i<=n; i++) {
        printf("%d,",aa[i]);
    }
}
```

输入:5

输出:1,3,4,5,7,8,10,13,14,16,17,19,21,23,

由于本题已知数据的有序性, 故可以使用二分法查找.

§5.7 生成菲波纳切数列

已知函数

$$F_1 = 1, F_2 = 1, F_n = F_{n-2} + F_{n-1}, n \geq 3$$

输出数列的前 12 项:

1,1,2,3,5,8,13,21,34,55,89,144

程序名称:C05_050.c

```
#include "stdio.h"
```

```
#include "stdlib.h"
```

```
void main() {
    int i;
    int F[13]={0,1,1};

    for (i=3; i<=12; i++) {
        F[i]=F[i-2]+F[i-1];
    }
    for (i=1; i<=12; i++) {
        printf("%d,",F[i]);
    }
}
```

结果为:1,1,2,3,5,8,13,21,34,55,89,144,

§5.8 分离十进制整数 a 的每一位数字

比如:

输入 12345 系统输出 1,2,3,4,5

输入 3456789 系统输出 3,4,5,6,7,8,9

程序名称:C05_060.c

```
#include "stdio.h"
```

```
#include "stdlib.h"
```

```
void main() {
    int i,x;
    int FF[15]={0};

    scanf("%d",&x);
    for (i=0; x>0; i++) {
        FF[i]=x%10;
        x/=10;
    }
    for (i--; i>=0; i--) {
        printf("%d,",FF[i]);
    }
}
```

输入:12345

输出:1,2,3,4,5,

§5.9 成绩均值与方差

已知学生成绩存于数组中:

81,83,84,77,78,80,93,94,88,60

编写程序, 求其成绩均值、方差.

程序名称:C05_070.c

```
#include "stdio.h"
```

```
#include "stdlib.h"
```

```
int main() {
    int i;
    int aa[10]={81,83,84,77,78,80,93,94,88,60};
    double sum,ave,dev;

    sum=0.0;
    for (i=0; i<10; i++) sum+=aa[i];
    ave=sum/10;
    dev=0.0;
    for (i=0; i<10; i++) {
        dev+=(aa[i]-ave)*(aa[i]-ave);
    }
    dev/=10;
    printf("ave=%lf, dev=%lf",ave,dev);
}
```

结果为:ave=81.800000, dev=83.560000

§5.10 运用数组打印空心图形

```

*****
*****
****
***
**
*
**
***
****
*****
*****

```

```

程序名称:C05_080.c
#include "stdio.h"
#include "stdlib.h"
void main() {
    int i;
    char str[12]="*****";
    printf("%s\n",str);
    for (i=0; i<=4; i++) {
        str[5-i]=' ';   str[5+i]=' ';
        printf("%s\n",str);
    }
    for (i=4; i>=0; i--) {
        str[5-i]='*';   str[5+i]='*';
        printf("%s\n",str);
    }
}

```

§5.11 冒泡法排序

已知学生成绩存于数组中:

81, 83, 84, 77, 78, 80, 93, 94, 88, 60

编写程序, 从小到大运用冒泡法排序.

```

程序名称:C05_090.c
#include "stdio.h"
#include "stdlib.h"
void main() {
    int i,j,k;
    int aa[10]={81,83,84,77,78,80,93,94,88,60};
    for (i=0; i<10; i++) {
        for (j=0; j<9-i; j++) {
            if (aa[j]>aa[j+1]) {
                k=aa[j]; aa[j]=aa[j+1]; aa[j+1]=k;
            }
        }
    }
    for (i=0; i<10; i++) printf("%d,",aa[i]);
}
结果为:60,77,78,80,81,83,84,88,93,94,

```

§5.12 选择排序

已知学生成绩存于数组中:

81, 83, 84, 77, 78, 80, 93, 94, 88, 60

编写程序, 从小到大运用选择排序.

```

程序名称:C05_100.c
#include "stdio.h"
#include "stdlib.h"

void main() {
    int i,j,n,k;
    int aa[10]={81,83,84,77,78,80,93,94,88,60};

    for (i=0; i<10; i++) {
        for (j=n=i; j<10; j++) {
            if (aa[n]>aa[j]) n=j; //记录最小者位置
        }
        if (n>i) {
            k=aa[i]; aa[i]=aa[n]; aa[n]=k;
        }
    }
    for (i=0; i<10; i++) printf("%d,",aa[i]);
}

```

§5.13 集合的交集

求两个数组的公共元素.

A={60,77,78,80,81,83,84,88,93,94}

B={51,58,60,78,83,84,95,99}

```

程序名称:C05_110.c
#include "stdio.h"
#include "stdlib.h"

void main() {
    int A[10]={60,77,78,80,81,83,84,88,93,94};
    int B[10]={51,58,60,78,83,84,95,99};
    int C[10]={0};
    int nA=10,nB=8,nC=-1,i,j;

    for (i=0,j=0; i<nA && j<nB; ) {
        if (A[i]==B[j]) {
            C[++nC]=A[i]; i++; j++;
        } else if (A[i]<B[j]) i++;
        else j++;
    }
    for (i=0; i<=nC; i++) printf("%d,",C[i]);
}
结果为:60,78,83,84,

```

§5.14 作业

1. 已知数组:

1, 3, 4, 7, 8, 10, 13, 14, 16, 17, 19, 21, 23, 29, 30

编写程序, 将上述数组中的首元素移动到数组的尾部, 得到新的数组:

3, 4, 7, 8, 10, 13, 14, 16, 17, 19, 21, 23, 29, 30, 1

2. 已知数组:

1, 3, 4, 7, 8, 10, 13, 14, 16, 17, 19, 21, 23, 29, 30

编写程序, 将上述数组中的元素倒序, 得到新的数组:

30, 29, 23, 21, 19, 17, 16, 14, 13, 10, 8, 7, 4, 3, 1

3. 生成卢卡斯序列:

$$\begin{cases} F_1 = 1, & F_2 = 2, & F_3 = 3 \\ F_n = F_{n-3} + F_{n-2} + F_{n-1}, & n \geq 4 \end{cases}$$

输出数列的前 12 项.

4. 插入法排序

已知学生成绩存于数组中:

81, 83, 84, 77, 78, 80, 93, 94, 88, 60

编写程序, 从小到大运用插入法排序.

提示: 仔细阅读《数组元素插入》外套一个循环即可实现.

5. 二分法查找数组中的元素

已知数组具有从小到大的顺序关系:

1, 3, 3, 3, 7, 7, 13, 14, 15, 17, 19, 21, 23, 29, 30

键盘输入一个数 x, 判断其是否在数组中.

例如: 输入 0, 输出 "不在数组中"

输入 1, 输出 "在数组中"

输入 4, 输出 "不在数组中"

输入 14, 输出 "在数组中"

输入 34, 输出 "不在数组中"

6. 用筛法求 100 以内的全部素数

提示: 首先定义一个数组 aa[101], 然后赋初值 aa[i] = i:

0, 1, 2, 3, 4, 5, ... 100

下面开始划去非素数, 方法是:

二重循环, 外层 i: 2 → 10 依次检查 aa[i] 是否为 0, 若是, 则继续检查下一个, 否则循环去掉 aa[i] 的倍数, 去掉的方法是将该值赋值为 0 即可.

原序列:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, ...

第一次处理(去掉2的倍数):

0, 1, 2, 3, 0, 5, 0, 7, 0, 9, 0, 11, 0, 13, 0, 15, 0, ...

第二次处理(去掉3的倍数):

0, 1, 2, 3, 0, 5, 0, 7, 0, 0, 0, 11, 0, 13, 0, 0, 0, ...

第二次处理(去掉5的倍数):

0, 1, 2, 3, 0, 5, 0, 7, 0, 0, 0, 11, 0, 13, 0, 0, 0, ...

... ..

最后输出大于 1 的非零整数即为所求.

7. 杨辉三角

要求产生如下的图形:

```

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1

```

要求产生 10 行.

提示: 定义一个一维数组 aa[12] = {1}, 其首项为 1, 其它项为 0, 执行如下的操作即可:

设当前状态为:

```

1 4 6 4 1

```

从后向前, 将前项加到后项上, 如

```

1 4 6 4 1-->1

```

```

1 4 6 4-->5 1

```

```

1 4 6-->10 5 1

```

```

1 4-->10 10 5 1

```

```

1-->5 10 10 5 1

```

输出结果:

```

1 5 10 10 5 1

```

8. 设有 10 个学生成绩为 {98, 99, 52, 62, 77, 89, 88, 42, 73, 78}, 编程序求出这些成绩中优秀 (90 100), 良好 (80 89), 中等 (70 79), 及格 (60 69), 不及格 (59 分以下) 各分数段的人数.

第六章 函数调用

§6.1 函数定义

返回值类型 函数名称(参数列表) {
 函数体
}

在C语言中,一切子程序均用函数完成.

§6.2 最小素因子函数

编写一个函数,存放在外部文件中,以便于主程序调用它.

程序名称:C06_010.h

```
int MinZYS(int x) {  
    int i;  
    for (i=2; i<x; i++) {  
        if (x%i==0) break;  
    }  
    return i;  
}
```

输入:91
输出:7

将外部编写好的函数包含进主文件中.

程序名称:C06_010.c

```
#include "stdio.h"  
#include "stdlib.h"  
#include "C06_010.h"  
  
void main() {  
    int k,m;  
    scanf("%d",&m);  
    k=MinZYS(m);  
    printf("%d\n",k);  
}
```

§6.3 正整数的质因数分解

比如:

输入 11 输出 11=11

输入 12 输出 12=2*2*3

程序名称:C06_020.c

```
#include "stdio.h"  
#include "stdlib.h"  
#include "C06_010.h"  
  
void main() {  
    int k,m,n;  
    scanf("%d",&m);
```

```
    printf("%d=",m);  
    for (n=m; n>1; ) {  
        k=MinZYS(n);  
        if (m>n) printf("*");  
        printf("%d",k);  
        n/=k;  
    }  
}
```

输入:60
输出:60=2*2*3*5

§6.4 最大值函数

程序名称:C06_030.h

```
int Max(int x,int y) {  
    return(x>y ? x:y);  
}
```

程序名称:C06_030.c

```
#include "stdio.h"  
#include "stdlib.h"  
#include "C06_030.h"  
  
void main() {  
    int m,n;  
    m=3; n=4;  
    printf("%d\n",Max(m,n));  
}
```

结果为:4

§6.5 最大公因子函数

程序名称:C06_040.h

```
int GYZ(int x,int y) {  
    x=x<0 ? -x:x; y=y<0 ? -y:y;  
    for ( ; x>0 && y>0; ) {  
        if (x>y) x%=y; else y%=x;  
    }  
    return(x+y);  
}
```

程序名称:C06_040.c

```
#include "stdio.h"  
#include "stdlib.h"  
#include "C06_040.h"  
  
void main() {  
    int m,n;  
    m=72; n=48;  
    printf("%d\n",GYZ(m,n));  
}
```

结果为:24

§6.6 随机数发生器函数

下面是一个简单的均匀随机数发生器,产生的是(0,1)区间上的均匀随机数,可以用这个发生器做很多事情,如估计定积分的大小、生成随机试卷等。

程序名称:C06_050.h

```
double Rand() {
    static unsigned long x=1;
    x=(x*314159269+453806245)%2147483648;
    return(x/2147483648.0);
}
```

设 $y = f(x) = x^2$, 计算定积分:

$$\int_0^1 x^2 dx$$

方法是在边长为1的正方形内均匀投点,该定积分就是落在 $y = f(x) = x^2$ 曲线下的阴影面积,也就是落在阴影部分的点数占总点数的比例。

程序名称:C06_050.c

```
#include "stdio.h"
#include "stdlib.h"
#include "C06_050.h"

void main() {
    long i,k;
    double x,y;
    for (i=0,k=0; i<2000000L; i++) {
        x=Rand(); y=Rand();
        if (x*x>y) k++;
    }
    printf("%lf\n",k/2000000.0);
}
```

结果为:0.333789

§6.7 组合函数

输出从0,1,2,3,4五个数中任意选取三个数的全部组合。

全部组合如下:

```
0 1 2
0 1 3
0 1 4
0 2 3
0 2 4
0 3 4
```

```
1 2 3
1 2 4
1 3 4
2 3 4
```

上面的情况很有规律,可以编写一个函数,从一个状态生成下一个状态。首先从后向前,找到第一个能够增加的位置,若不存在增加的位置,则说明已经全部结束了,若找到能够增加的位置,则增加1,然后在增加的位置后,数字按照升序排列。

程序名称:C06_060.h

```
long Combination(long n,long m) {
    int i,j;
    for (i=m-1;i>=0;i--) if(x[i]+m-i<n) break;
    if (i<0) return(1); /* over */
    for (x[i]++,j=i+1;j<m;j++) x[j]=x[j-1]+1;
    return (0); /* ok! */
}
```

程序名称:C06_060.c

```
#include "stdio.h"
#include "stdlib.h"
int x[10]; /* 只能对下标求组合 */
#include "C06_060.h"
void main() {
    int i,m,n;
    n=5; /* 待组合的元素总个数 */
    m=3; /* 组合数 */
    for (i=0; i<n; i++) x[i]=i; /* 初始化 */
    for ( ; ; ) {
        for (i=0; i<m; i++) printf(" %d",x[i]);
        printf("\n");
        if (Combination(n,m)) break;
    }
}
```

§6.8 计算今天星期几

已知2012年1月1日是星期日,编写一个函数,输入今天的日期,输出星期几。要求输入三个数,年,月,日,然后输出一个数0-6,其中0:星期日,1:星期一,...6:星期六。

程序名称:C06_070.h

```
long RunNian(long yyyy) { // 是否闰年
    if (yyyy%400==0) return(1);
    if (yyyy%100==0) return(0);
    if (yyyy%4==0) return(1);
    return(0);
}
```

```

long YueFen(long m) { // 该月天数
    if (m==4||m==6||m==9||m==11) return(30);
    if (m==2) return(28);
    return(31);
}
// 计算今天是星期几
long XingQi(long yyyy,long mm,long dd) {
    long nY,nM,nSum;
    for (nY=2012,nSum=0; nY<yyyy; nY++) {
        nSum += 365+RunNian(nY);
    }
    for (nM=1; nM<mm; nM++) {
        nSum += YueFen(nM);
        if (nM==2) nSum += RunNian(nY); // 闰年
    }
    nSum += dd;
    return ((nSum-1)%7); // 从零开始
}

```

程序名称:C06_070.c
#include "stdio.h"
#include "stdlib.h"
#include "C06_070.h"

```

void main() {
    int y,m,d;
    scanf("%d%d%d",&y,&m,&d);
    printf("%ld",XingQi(y,m,d));
}
输入:2018 9 4
输出:2

```

也可以用文字显示。

```

程序名称:C06_080.c
#include "stdio.h"
#include "stdlib.h"
#include "C06_070.h"
void main() {
    int y,m,d,k;
    scanf("%d%d%d",&y,&m,&d);
    k=XingQi(y,m,d);
    switch(k) {
        case 0: printf("星期日\n"); break;
        case 1: printf("星期一\n"); break;
        case 2: printf("星期二\n"); break;
        case 3: printf("星期三\n"); break;
        case 4: printf("星期四\n"); break;
        case 5: printf("星期五\n"); break;
        case 6: printf("星期六\n"); break;
    }
}
输入:2018 9 4
输出:星期二

```

§6.9 数组有序插入新值

数组具有从小到大的顺序关系:

4, 7, 8, 16, 17

将下列数存入上述数组中

3, 1, 10, 13, 19

要求保持原有顺序。

程序名称:C06_090.h

```

int Insert(int x) {
    int i;
    for (i=n; i>=0; i--) {
        if (aa[i]>x) aa[i+1]=aa[i]; else break;
    }
    aa[i+1]=x;    n++;
    return(0);
}

```

程序名称:C06_090.c

```

#include "stdio.h"
#include "stdlib.h"

```

```

int n,aa[30]={4,7,8,16,17};
#include "C06_080.h"

```

```

void main() {
    int i;
    n=4;
    Insert(3);
    for (i=0; i<=n; i++) printf("%d,",aa[i]);
    printf("\n");
    Insert(1);
    for (i=0; i<=n; i++) printf("%d,",aa[i]);
    printf("\n");
    Insert(10);
    for (i=0; i<=n; i++) printf("%d,",aa[i]);
    printf("\n");
    Insert(13);
    for (i=0; i<=n; i++) printf("%d,",aa[i]);
    printf("\n");
    Insert(19);
    for (i=0; i<=n; i++) printf("%d,",aa[i]);
    printf("\n");
}

```

结果为:

3,4,7,8,16,17,
1,3,4,7,8,16,17,
1,3,4,7,8,10,16,17,
1,3,4,7,8,10,13,16,17,
1,3,4,7,8,10,13,16,17,19,

§6.10 作业

1. 输出 100 以内的所有正整数的质因数分解.

```
1=1
2=2
3=3
4=2*2
5=5
6=2*3
7=7
8=2*2*2
9=3*3
...
96=2*2*2*2*2*3
97=97
98=2*7*7
99=3*3*11
100=2*2*5*5
```

2. 编写一个插入排序的函数.

全局变量为整型数组 A 及其元素个数 n, 函数的入口参数为待插入元素 x, 假设 n 小于数组的大小.

编写一个主程序, 调用上述子程序, 数据为 1,3,4,7,8,10,13,14,16; 键盘输入待插入数值.

3. 编写一个针对整型数组的二分法的查找函数.

全局变量为整型数组 A 及其元素个数 n, 函数的入口参数为待查询元素 x.

编写一个主程序, 调用上述子程序, 数据为 1,3,3,3,7,7,13,14,15,17,19,21,23,29,30. 键盘输入待查数据, 然后显示“在数组中”或“不在数组中”.

4. 编写产生杨辉三角的函数.

全局变量为整型数组 A, 数组的总大小为 n, 假设当前行状态已知, 编写函数生成下一行状态.

编写主程序调用上述子程序, 要求产生 10 行.

5. 利用随机函数产生 100 个 18-25 之间的数, 统计产生了几几个 18, 几个 19,..., 几个 25.

6. 用随机函数产生 52 个 1-52 之间互不相等的数, 存入数组 A[52] 中, 规定 1-13 为

黑桃,14-26 为红桃,27-39 为方块,40-52 为梅花, 试按下列格式打印出这 52 张牌: 花色, 值.(注: 如黑桃 10, 则花色为黑桃, 值为 10)

第七章 二维数组

§7.1 数组的定义

```
int    ab[4][3]; // 二维数组,4行3列
double bc[5][2];
int    kk[3][4][5]; // 三维数组
```

§7.2 数组的初始化

```
int  ab[2][3]={1,5,3},{6,8,9}};
int  ac[3][3]={0};
int  ac[3][3]={1,2},{1},{0,2}};
int  ac[2][3]={0,1,2,3,4,5};
int  kk[3][4][5]={1,5,3},{6},{6}};
```

§7.3 矩阵加法

已知矩阵 A,B, 求 A+B.

$$A = \begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 3 & 2 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 2 & 1 & 0 \\ 2 & 3 & 1 \\ 3 & 1 & 2 \end{bmatrix}$$

程序名称:C07_010.c

```
#include "stdio.h"
#include "stdlib.h"
```

```
void main() {
    int i,j,n;
    double A[3][3]={1,5,3},{4,2,1},{3,2,0}};
    double B[3][3]={2,1,0},{2,3,1},{3,1,2}};
    double C[3][3];

    n=3;
    for (i=0; i<n; i++) {
        for (j=0; j<n; j++) {
            C[i][j]=A[i][j]+B[i][j];
        }
    }
    for (i=0; i<n; i++) {
        for (j=0; j<n; j++) {
            printf(" %lf",C[i][j]);
        }
        printf("\n");
    }
}
```

§7.4 矩阵的消法变换

已知矩阵:

将第 3 行乘以 4, 加到第 1 行上去.

$$A = \begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 3 & 2 & 0 \end{bmatrix}$$

程序名称:C07_020.c

```
#include "stdio.h"
#include "stdlib.h"
```

```
void main() {
    int i,j,n;
    double A[3][3]={1,5,3},{4,2,1},{3,2,0}};

    n=3;
    for (j=0; j<n; j++) {
        A[0][j] += 4*A[2][j];
    }
    for (i=0; i<n; i++) {
        for (j=0; j<n; j++) {
            printf(" %lf",A[i][j]);
        }
        printf("\n");
    }
}
```

§7.5 学生成绩

现有数学、语文、外语、政治、物理、化学
六门课程成绩如下:

$$A = \begin{bmatrix} 81 & 75 & 83 & 81 & 75 & 83 \\ 44 & 92 & 71 & 53 & 82 & 80 \\ 93 & 52 & 80 & 54 & 92 & 71 \end{bmatrix}$$

1. 求理科成绩平均分 (数学、物理、化学).
2. 统计不及格的人数和门数.

程序名称:C07_030.c

```
#include "stdio.h"
#include "stdlib.h"
```

```
double aa[3][6]={
    {81, 75, 83, 81, 75, 83},
    {44, 92, 71, 53, 82, 80},
    {93, 52, 80, 54, 92, 71}
};
```

```
void main() {
    int i,j,k,rs,ms;
```

```

double s;

// 理科平均分
s=0.0;
for (i=0,k=0; i<3; i++,k+=3) {
    s += aa[i][0] + aa[i][4] + aa[i][5];
}
printf("平均分=%lf\n",s/k);
// 不及格人数
rs=0;
for (i=0; i<3; i++) {
    for (j=0; j<6; j++) {
        if (aa[i][j]<60) { rs++; break; }
    }
}
printf("人数=%d\n",rs);
// 不及格门数
ms=0;
for (j=0; j<6; j++) {
    for (i=0; i<3; i++) {
        if (aa[i][j]<60) { ms++; break; }
    }
}
printf("门数=%d\n",ms);
}

```

§7.6 获取子矩阵

获得一个六阶方阵的三行和三列交叉位置的元素构成的子矩阵。

程序名称:C07_040.c
#include "stdio.h"
#include "stdlib.h"

```

void main() {
    int i,ii[10]={1,3,5}; // 行号列表
    int j,jj[10]={1,2,6}; // 列号列表
    double AA[6][6]={ // 原始矩阵
        {0,1,2,3,4,5},
        {1,2,3,4,5,6},
        {2,3,4,5,6,7},
        {3,4,5,6,7,8},
        {4,5,6,7,8,9},
        {5,6,7,8,9,0}
    };
    double BB[3][3]={0}; // 目标矩阵

    for (i=0; ii[i]!=0; i++) {
        for (j=0; jj[j]!=0; j++) {
            BB[i][j]=AA[ii[i]-1][jj[j]-1];
        }
    }
}

```

```

for (i=0; i<3; i++) {
    for (j=0; j<3; j++) {
        printf("%lf ",BB[i][j]);
    }
    printf("\n");
}

```

结果为:

```

0.000000 1.000000 5.000000
2.000000 3.000000 7.000000
4.000000 5.000000 9.000000

```

§7.7 化向量组为单位向量组

矩阵的每一列为一个向量,需要对每一个向量进行单位化,使得每一个向量的所有分量的平方和为1。

程序名称:C07_050.c
#include "stdio.h"
#include "stdlib.h"
#include "math.h"

```

void main() {
    int i,j;
    double sum,AA[4][6]={
        {0,1,2,3,4,5},
        {1,2,3,4,5,6},
        {2,3,4,5,6,7},
        {3,4,5,6,7,8}
    };
    for (j=0; j<6; j++) {
        for (i=0,sum=0; i<4; i++) {
            sum += AA[i][j]*AA[i][j];
        }
        sum=sqrt(sum);
        for (i=0; i<4; i++) {
            AA[i][j] /= sum;
        }
    }
    for (i=0; i<4; i++) {
        for (j=0; j<6; j++) {
            printf("%.4lf ",AA[i][j]);
        }
        printf("\n");
    }
}

```

结果为:

```

0.0000 0.1826 0.2722 0.3235 0.3563 0.3790
0.2673 0.3651 0.4082 0.4313 0.4454 0.4549
0.5345 0.5477 0.5443 0.5392 0.5345 0.5307
0.8018 0.7303 0.6804 0.6470 0.6236 0.6065

```


§7.8 打印图形

图形如下: 要求先定义二维数组, 然后添数, 最后将整个数组打印.

```
1  2  3  4  5  6  7
24 25 26 27 28 29  8
23 40 41 42 43 30  9
22 39 48 49 44 31 10
21 38 47 46 45 32 11
20 37 36 35 34 33 12
19 18 17 16 15 14 13
```

程序名称:C07_060.c

```
#include "stdio.h"
```

```
#include "stdlib.h"
```

```
#define N 7
```

```
void main() {
    int i,j,k,n;
    int A[N][N]={0};

    for (n=0,k=1; n<(N+1)/2; n++) {
        i=j=n; A[i][j]=k;
        for (; j<N-n-1; j++,k++) A[i][j]=k;
        for (; i<N-n-1; i++,k++) A[i][j]=k;
        for (; j>n; j--,k++) A[i][j]=k;
        for (; i>n; i--,k++) A[i][j]=k;
    }

    for (i=0; i<N; i++) {
        for (j=0; j<N; j++) {
            printf(" %2d",A[i][j]);
        }
        printf("\n");
    }
}
```

§7.9 奇数阶幻方

奇数阶幻方的一种构造方法如下: 首先在最后一行的中间位置填写 1, 然后在下一行下一列位置填写 2, 如果行向下超界则转到首行, 若列向后超界, 则转到首列, 若下一行下一列位置已经填写过数了, 则在当前位置的上一行当前列位置继续填写.

程序名称:C07_070.c

```
#include "stdio.h"
```

```
#include "stdlib.h"
```

```
#define N 5
```

```
void main() {
    int i,j,k,x,y,A[N][N]={0};
```

```
    for (i=N-1,j=N/2,k=1; k<=N*N; k++) {
        A[i][j]=k; x=(i+1)%N; y=(j+1)%N;
        if (A[x][y]==0) {
            i=x; j=y;
        } else i=(i-1+N)%N;
    }
    for (i=0; i<N; i++) {
        for (j=0; j<N; j++) {
            printf("%2d ",A[i][j]);
        }
        printf("\n");
    }
}
```

结果为:(N=3)

结果为:(N=5)

```
4  9  2      11 18 25  2  9
3  5  7      10 12 19 21  3
8  1  6      4  6 13 20 22
              23  5  7 14 16
              17 24  1  8 15
```

验证 5 阶幻方的每一行和每一列以及对角线的元素之和是否均相同.

程序名称:C07_080.c

```
#include "stdio.h"
```

```
#include "stdlib.h"
```

```
#define N 5
```

```
void main() {
    int i,j,k,x,y,A[N][N]={0};
    int sum,s1,s2;
    for (i=N-1,j=N/2,k=1; k<=N*N; k++) {
        A[i][j]=k; x=(i+1)%N; y=(j+1)%N;
        if (A[x][y]==0) {
            i=x; j=y;
        } else i=(i-1+N)%N;
    }
    for (i=0; i<N; i++) {
        for (j=0,sum=0; j<N; j++) sum += A[i][j];
        printf("%d,",sum);
    }
    printf("\n");
    for (j=0; j<N; j++) {
        for (i=0,sum=0; i<N; i++) sum += A[i][j];
        printf("%d,",sum);
    }
    printf("\n");
    for (i=0,s1=0; i<N; i++) s1 += A[i][i];
    for (i=0,s2=0; i<N; i++) s2 += A[i][N-i-1];
    printf("%d,%d\n",s1,s2);
}
```

结果为:

```
65,65,65,65,65,
65,65,65,65,65,
65,65
```

§7.10 作业

1. 矩阵乘法, 求解 AB:

$$A = \begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 3 & 2 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 2 & 1 & 0 \\ 2 & 3 & 1 \\ 3 & 1 & 2 \end{bmatrix}$$

2. 矩阵的换法变换

交换矩阵的第 1 行和第 3 行, 其中矩阵如下:

$$A = \begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 3 & 2 & 0 \end{bmatrix}$$

3. 矩阵的所有元素之和

实数矩阵如下:

$$A = \begin{bmatrix} 1.1 & 5 & 3 \\ 4.2 & 2 & 1 \\ 3.3 & 2 & 0 \end{bmatrix}$$

4. 用二维数组存储杨辉三角, 然后打印.

要求产生如下的图形:

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
```

要求产生10行.

5. 用二维数组存储图形, 然后打印.

图形如下:(三对角矩阵)

```
1 1 0 0 0 0 0
1 2 1 0 0 0 0
0 3 3 1 0 0 0
0 0 5 4 1 0 0
0 0 0 7 5 1 0
0 0 0 0 9 6 1
0 0 0 0 0 11 7
```

6. 二维数组存储图形并打印.(蛇行三角)

```
22 0 0 0 0 0 0
16 23 0 0 0 0 0
11 17 24 0 0 0 0
7 12 18 25 0 0 0
4 8 13 19 26 0 0
2 5 9 14 20 27 0
1 3 6 10 15 21 28
```

第八章 堆栈技术

利用一维数组来构造一种数据存储形式，通过形象的动作序列来表现数据的处理过程。

§8.1 堆栈的一般形式

```
int ss[15],sp; // 堆栈定义
... ..
sp=-1; // 初始化
... ..
ss[++sp]=k; // 变量k的结果压栈
... ..
k=ss[sp--]; // 出栈,存入变量k中
... ..
```

§8.2 十进制转换为二进制

例如: 44 --> 101100
27 --> 11011

程序名称:C08_010.c
#include "stdio.h"
#include "stdlib.h"

```
void main() {
    int k;
    int ss[15],sp; // 堆栈定义
    scanf("%d",&k);
    for (sp=-1; k>0; k/=2) {
        ss[++sp] = k%2; // 变量k的结果压栈
    }
    for ( ; sp>=0; ) printf("%d",ss[sp--]);
    printf("\n");
}
```

输入:44
输出:101100

二进制转换为十进制程序.

程序名称:C08_020.c
#include "stdio.h"
#include "stdlib.h"

```
void main() {
    char ch;
    int k=0;
    while ((ch=getchar())!='\n') {
        k=2*k+ch-'0';
    }
    printf("%d\n",k);
}
```

输入:101100
输出:44

§8.3 回文数

一个 5 位数，判断它是不是回文数。即 12321 是回文数，个位与万位相同，十位与千位相同。

程序名称:C08_030.c

#include "stdio.h"
#include "stdlib.h"

```
void main() {
    int k;
    int ss[15],sp; // 堆栈定义
    scanf("%d",&k);
    for (sp=-1; k>0; k/=10) {
        ss[++sp] = k%10; // 变量k的结果压栈
    }
    if (ss[0]==ss[4] && ss[1]==ss[3]) {
        printf("Yes");
    } else printf("No");
}
```

输入:23432
输出:Yes

§8.4 质因数分解

例如: 100=2*2*5*5

程序名称:C08_040.c

#include "stdio.h"
#include "stdlib.h"

```
void main() {
    int i,k;
    int ss[15],sp; // 堆栈定义

    scanf("%d",&k);
    printf("%d=",k);
    for (sp=-1; k>1; k/=i) {
        for (i=2; i<k; i++) {
            if (k%i==0) break;
        }
        ss[++sp] = i; // 变量i压栈
    }
    printf("%d",ss[0]);
    for (i=1; i<=sp; i++) printf(" *%d",ss[i]);
    printf("\n");
}
```

输入:100
输出:100=2*2*5*5

§8.5 字符串反序

利用堆栈缓存, 然后倒序输出.

```

程序名称:C08_050.c
#include "stdio.h"
#include "stdlib.h"

void main() {
    int st[200],sp; // 堆栈定义
    char k;

    for (sp=-1; (k=getchar())!='\n'; ) {
        st[++sp]=k;
    }
    for ( ; sp>=0; ) printf("%c",st[sp--]);
}
输入:12345
输出:54321

```

§8.6 双侧堆栈

相当于两个堆栈, 使用一个数组两个指针, 一个从前向后存储, 一个从后向前存储.

要求将数组中小于等于 5 的数排在左侧, 比 5 大的排在右侧.

数组: 2,1,7,3,4,9,3,8,4,6
 边界数: 5
 结果: 4,3,4,3,1,2,6,8,9,7

```

程序名称:C08_060.c
#include "stdio.h"
#include "stdlib.h"

void main() {
    int st[200],sp1,sp2; // 堆栈定义
    int aa[10]={2,1,7,3,4, 9,3,8,4,6};
    int i,k;

    sp1=-1; sp2=200;
    k=5;
    for (i=0; i<10; i++) {
        if (aa[i]<=k) st[++sp1]=aa[i];
        else st[--sp2]=aa[i];
    }
    for ( ; sp1>=0; ) printf("%d,",st[sp1--]);
    for ( ; sp2<200; ) printf("%d,",st[sp2++]);
}
结果为:4,3,4,3,1,2,6,8,9,7,

```

§8.7 复杂连分数计算

设复杂连分数:

$$\frac{1}{2 + \frac{3}{4 + \frac{5}{6}}} \quad \text{记为: } \frac{1}{2} + \frac{3}{4} + \frac{5}{6}$$

一般连分数记为:

$$\frac{a_1}{b_1} + \frac{a_2}{b_2} + \cdots + \frac{a_n}{b_n}$$

编程求解:

$$\frac{1}{b_1} + \frac{1}{b_2} + \cdots + \frac{1}{b_n}$$

其中 b_1, b_2, \dots, b_n 均为非零实数, 要求最后增加输入一个 0, 它不参与计算, 仅表示输入结束.

例如: 输入 1 2 3 0 输出 0.700000

```

程序名称:C08_070.c
#include "stdio.h"
#include "stdlib.h"

void main() {
    double ss[10]; // 堆栈定义
    int sp; // 堆栈指针
    double k,jg; // 中间缓存,结果

    for (sp=-1; ; ) {
        scanf("%lf",&k);
        if (k==0) break;
        ss[++sp] = k; // 变量k的结果压栈
    }

    for (jg=0; sp>=0; ) {
        jg=1.0/(ss[sp--]+jg);
    }
    printf("%lf\n",jg);
}

```

连分数的应用非常广泛, 任何实数都可以使用连分数逼近, 如:

$$\frac{\sqrt{5}-1}{2} = \frac{1}{1} + \frac{1}{1} + \frac{1}{1} + \cdots$$

$$\sqrt{2}-1 = \frac{1}{2} + \frac{1}{2} + \frac{1}{2} + \cdots$$

§8.8 数组数据翻转

设一个有 10 个元素的数组, 现将前后数据翻转. 这里想介绍一个常规方法, 利用两个指针, 从前向后和从后向前, 同时向中间推进, 在行进过程中交换对应数据, 直到两个指针相遇时终止.

原数组:1,2,3,4,5,9,8,7,6,5

翻转后:5,6,7,8,9,5,4,3,2,1

程序名称:C08_080.c

#include "stdio.h"

#include "stdlib.h"

```
void main() {
    int p1,p2; // 指针
    int aa[10]={1,2,3,4,5,9,8,7,6,5};
    int i,k;

    for (p1=0,p2=9; p1<p2; p1++,p2--) {
        k=aa[p1]; aa[p1]=aa[p2]; aa[p2]=k;
    }
    for (i=0; i<=9; i++) printf("%d,",aa[i]);
}
结果为:5,6,7,8,9,5,4,3,2,1,
```

§8.9 利用堆栈走迷宫

现有 9*9 的一块地,1 代表墙壁,0 代表地板,某人从 0,0 位置进入,从 8,8 位置走出,轨迹用 2 表示. 设某人在 (x,y) 位置,显然他有四个可能移动的方向,(1,0),(-1,0),(0,1),(0,-1). 用 ss[] 存储正在搜索的方向,共有四个方向:0,1,2,3. 用 xx[] 和 yy[] 存储当前的位置.

程序名称:C08_090.c

#include "stdio.h"

#include "stdlib.h"

```
int jz[9][9]={ //1代表墙壁,0代表地板
    {0, 1, 1, 1, 1, 1, 1, 1, 1},
    {0, 0, 0, 0, 0, 0, 1, 0, 1},
    {1, 0, 1, 1, 0, 1, 1, 0, 1},
    {1, 0, 0, 1, 0, 0, 1, 0, 1},
    {1, 1, 0, 1, 1, 0, 0, 0, 1},
    {1, 0, 0, 1, 0, 1, 0, 0, 1},
    {1, 1, 0, 1, 0, 1, 1, 1, 1},
    {1, 0, 0, 0, 0, 0, 0, 0, 0},
    {1, 1, 1, 1, 1, 1, 1, 1, 0}
};
```

```
void main() {
    int i,j,x,y;
    int xx[100],yy[100],ss[100],sp; //堆栈
    int Fx[4]={1,-1,0, 0}; // x方向
    int Fy[4]={0, 0,1,-1}; // y方向

    x=0; y=0; sp=0; ss[sp]=0;
    xx[sp]=x; yy[sp]=y; jz[x][y]=2;
    for ( ; !(x==8 && y==8); ) { //0,0入,8,8出
        i=ss[sp]; //得到当前栈顶内容
        if (i>3) { // 无路可走了
            jz[x][y]=0; // 恢复当前值
            --sp; ss[sp]++; // 后退
            x=xx[sp]; y=yy[sp]; //上步的x,y坐标
        } else { //先判断超不超界
            if (x+Fx[i]>=0 && x+Fx[i]<=8 &&
                y+Fy[i]>=0 && y+Fy[i]<=8) {
                if (jz[x+Fx[i]][y+Fy[i]]==0) {
                    ss[++sp]=0; // 可以到达新位置
                    x+=Fx[i]; y+=Fy[i]; //定义新位置
                    xx[sp]=x; yy[sp]=y; //存储新位置
                    jz[x][y]=2; // 设置已经走过了
                } else { // 此位置是墙壁或已经走过
                    ss[sp]++; // 测试下一种情况
                }
            } else { // 超界,换下一种情况继续测试
                ss[sp]++; // 测试下一种情况
            }
        }
    }
    for (i=0; i<=8; i++) { //输出矩阵,2就是轨迹
        for (j=0; j<=8; j++) {
            printf("%2d",jz[i][j]);
        }
        printf("\n");
    }
}
```

结果为:

```
2 1 1 1 1 1 1 1 1
2 2 0 0 0 0 1 0 1
1 2 1 1 0 1 1 0 1
1 2 2 1 0 0 1 0 1
1 1 2 1 1 0 0 0 1
1 0 2 1 0 1 0 0 1
1 1 2 1 0 1 1 1 1
1 0 2 2 2 2 2 2 2
1 1 1 1 1 1 1 1 2
```

§8.10 作业

1. 十进制转换为十六进制

例如: 输入 123 输出 7B

输入 98765 输出 181CD

输入 12345789 输出 75BCD15

2. 数位分离

一个 $2n$ 位数, 将其分离成奇数位在前, 偶数位在后的一个新数.

例如: 123456 --> 135246

36235789 --> 32586379

3. 利用堆栈进行图形填充

已知图形, 给定种子点坐标, 要求将图形中所有与种子点连通的所有点都标记为 2. 图形如下:

```
{1, 1, 1, 1, 1, 1, 1, 1, 1},
{1, 0, 0, 0, 1, 0, 0, 0, 1},
{1, 0, 1, 1, 0, 1, 1, 0, 1},
{1, 0, 0, 1, 0, 0, 1, 0, 1},
{1, 1, 0, 1, 1, 0, 0, 0, 1},
{1, 0, 0, 1, 0, 0, 0, 0, 1},
{1, 1, 0, 1, 0, 1, 1, 0, 1},
{1, 0, 0, 0, 1, 0, 0, 0, 1},
{1, 1, 1, 1, 1, 1, 1, 1, 1}
```

例如: 当给出种子点坐标 (1,1) 时, 将与 (1,1) 点连通的上下左右点都标记为 2, 如下图.

```
{1, 1, 1, 1, 1, 1, 1, 1, 1},
{1, 2, 2, 2, 1, 0, 0, 0, 1},
{1, 2, 1, 1, 0, 1, 1, 0, 1},
{1, 2, 2, 1, 0, 0, 1, 0, 1},
{1, 1, 2, 1, 1, 0, 0, 0, 1},
{1, 2, 2, 1, 0, 0, 0, 0, 1},
{1, 1, 2, 1, 0, 1, 1, 0, 1},
{1, 2, 2, 2, 1, 0, 0, 0, 1},
{1, 1, 1, 1, 1, 1, 1, 1, 1}
```

当给出种子点坐标 (1,7) 时, 如下图.

```
{1, 1, 1, 1, 1, 1, 1, 1, 1},
{1, 0, 0, 0, 1, 2, 2, 2, 1},
{1, 0, 1, 1, 2, 1, 1, 2, 1},
{1, 0, 0, 1, 2, 2, 1, 2, 1},
{1, 1, 0, 1, 1, 2, 2, 2, 1},
{1, 0, 0, 1, 2, 2, 2, 2, 1},
{1, 1, 0, 1, 2, 1, 1, 2, 1},
{1, 0, 0, 0, 1, 2, 2, 2, 1},
{1, 1, 1, 1, 1, 1, 1, 1, 1}
```

当给出种子点坐标 (2,5) 时, 如下图.

```
{1, 1, 1, 1, 1, 1, 1, 1, 1},
{1, 0, 0, 0, 1, 0, 0, 0, 1},
{1, 0, 1, 1, 0, 2, 2, 0, 1},
{1, 0, 0, 1, 0, 0, 2, 0, 1},
{1, 1, 0, 1, 1, 0, 0, 0, 1},
{1, 0, 0, 1, 0, 0, 0, 0, 1},
{1, 1, 0, 1, 0, 1, 1, 0, 1},
{1, 0, 0, 0, 1, 0, 0, 0, 1},
{1, 1, 1, 1, 1, 1, 1, 1, 1}
```

4. 阶乘的质因数分解

求 $n!$, 其中 $10 < n < 50$, 要求相同质因数合并用 \wedge 来表示, 例如:

例如: $13! = 2^10 * 3^5 * 5^2 * 7^1 * 11^1 * 13^1$

注意: 不要将 $n!$ 计算出来, 否则可能会超出整型数值的表示范围.

提示: 显然, 13 以内的所有素数都会在分解中出现, 而且 13 中含有 2 的因子个数为

$$[13/2] + [13/4] + [13/8] + [13/16] + [13/32] + \dots$$

$$= 6 + 3 + 1 + 0 + 0 + \dots = 10$$

第九章 函数递归

函数内直接或间接调用自己本身,称之为递归调用. 思维简单编程快.

若求一个较大数的运算可以转化为求较小数的运算, 用同样的方法再求更小的数, 反复调用自己, 但是每一次的参数不同, 参数逐渐减小, 直至可求.

§9.1 十进制转换为二进制

例如: 44 --> 101100

程序名称:C09_010.c

```
#include "stdio.h"
```

```
#include "stdlib.h"
```

```
void JZ(int n) {
    if (n<=1) printf("%d",n);
    else {
        JZ(n/2);
        printf("%d",n%2);
    }
}
```

```
void main() {
    int k;
    scanf("%d",&k);
    JZ(k);
}
```

输入:44

输出:101100

§9.2 菲波纳切数列的递归计算

数列: 1,1,2,3,5,8,...

公式: $F(n+2)=F(n+1)+F(n)$, $n>0$

求 $F(12)$.

程序名称:C09_020.c

```
#include "stdio.h"
```

```
#include "stdlib.h"
```

```
int F(int n) {
    if (n<=2) return(1);
    return(F(n-1)+F(n-2));
}
```

```
void main() {
    printf("%d\n",F(12));
}
```

结果为:144

§9.3 递归计算杨辉三角

计算杨辉三角的第 6 行.

公式: $F(m,1)=1$, $m>0$

公式: $F(m,n)=0$, $n>m$

公式: $F(m,n)=F(m-1,n-1)+F(m-1,n)$

数列: 1 5 10 10 5 1

程序名称:C09_030.c

```
#include "stdio.h"
```

```
#include "stdlib.h"
```

```
int YH(int m,int n) {
    if (n==1) return(1);
    if (n>m) return(0);
    return(YH(m-1,n-1)+YH(m-1,n));
}
```

```
void main() {
    int i;
    for (i=1; i<=6; i++) { // 第6行有6个元素
        printf("%4d",YH(6,i));
    }
}
```

结果为: 1 5 10 10 5 1

§9.4 伞型序递归

一个正整数 n , 不用循环和其他变量, 按照 $n, 2n, 4n, 8n, \dots$ 的顺序递增, 当值大于 5000 时, 把值按照原序和反序输出来.

例: $n=1237$, 则输出为:

1237, 2474, 4948, 9896, 9896, 4948, 2474, 1237,

程序名称:C09_040.c

```
#include "stdio.h"
```

```
#include "stdlib.h"
```

```
void Out(int m) {
    printf("%d,",m);
    if (m<5000) Out(2*m);
    printf("%d",m);
}
```

```
void main() {
    int n;
    scanf("%d",&n);
    Out(n);
}
```

§9.5 利用递归求最大公因子

```

程序名称:C09_050.c
#include "stdio.h"
#include "stdlib.h"

int gyz(int a,int b) {
    if (a==0 || b==0) return(a+b);
    return(a>b ? gyz(a%b,b) : gyz(b%a,a));
}

void main() {
    printf("%d\n",gyz(72,48));
}

```

结果为:24

§9.6 对策局势的确定

一堆火柴 15 根, 甲乙二人轮流拿, 要求每次最多拿三根, 最少拿一根, 谁最后拿没谁赢, 设甲先拿, 问对甲而言此局势是赢局还是输局?

```

程序名称:C09_060.c
#include "stdio.h"
#include "stdlib.h"

int JS(int n) {
    int i;
    if (n>=1 && n<=3) return(1); // 赢
    for (i=1; i<=3; i++) {
        if (JS(n-i)==0) return(1);
    }
    return(0);
}

void main() {
    printf("%d\n",JS(15));
}

```

结果为:1

§9.7 利用递归实现连分数计算

编程求解:

$$\frac{1}{b_1} + \frac{1}{b_2} + \dots + \frac{1}{b_n}$$

其中 b_1, b_2, \dots, b_n 均为非零实数, 要求最后增加输入一个 0, 它不参与计算, 仅表示输入结束。

例如: 输入 1 2 3 0 输出 0.700000

```

程序名称:C09_070.c
#include "stdio.h"
#include "stdlib.h"
double JS() {
    double k;
    scanf("%lf",&k);
    if (k==0) return(0);
    return(1.0/(k+JS()));
}
void main() {
    printf("%lf\n",JS());
}

```

§9.8 递归求根

函数:

$$f(x) = x^3 - 5x + 1, f(0) > 0, f(1) < 0,$$

求 $f(x) = 0$ 在 $(0,1)$ 上的一个实根。

```

程序名称:C09_080.c
#include "stdio.h"
#include "stdlib.h"

double gen(double a,double b) {
    double c;
    c=(a+b)/2.0;
    if (b-a<0.001) return(c);
    if (c*c*c-5*c+1>0) return(gen(c,b));
    return(gen(a,c));
}

void main() {
    printf("%lf\n",gen(0,1));
}

```

结果为:0.201660

非递归二分法求根。

```

程序名称:C09_090.c
#include "stdio.h"
#include "stdlib.h"

void main() {
    double a,b,c;

    a=0; b=1.0;
    while (b-a>0.001) {
        c=(a+b)/2.0;
        if (c*c*c-5*c+1>0) a=c; else b=c;
    }
    printf("%lf\n", (a+b)/2.0);
}

```


§9.9 递归实现全排列

全排列(压缩排版)

```
1,2,3,4, 2,1,3,4, 3,1,2,4, 4,1,2,3,
1,2,4,3, 2,1,4,3, 3,1,4,2, 4,1,3,2,
1,3,2,4, 2,3,1,4, 3,2,1,4, 4,2,1,3,
1,3,4,2, 2,3,4,1, 3,2,4,1, 4,2,3,1,
1,4,2,3, 2,4,1,3, 3,4,1,2, 4,3,1,2,
1,4,3,2, 2,4,3,1, 3,4,2,1, 4,3,2,1,
```

程序名称:C09_100.c

#include "stdio.h"

#include "stdlib.h"

int a[10],m=3; // 4个数的全排列

```
void Out(int n) {
    int i,j,s;
    if (n==m) {
        for (i=0; i<=m; i++) printf("%d,",a[i]);
        printf("\n");
        return;
    }
    for (i=n; i<=m; i++) {
        s=a[i]; // 更新
        for (j=i; j>n; j--) a[j]=a[j-1];
        a[n]=s; // 向后循环移动
        Out(n+1); // 递归
        s=a[n]; // 恢复
        for (j=n; j<i; j++) a[j]=a[j+1];
        a[i]=s; // 向前循环移动
    }
}
```

```
void main() {
    int i;
    for (i=0; i<=m; i++) a[i]=i+1;
    Out(0);
}
```

下面是非递归实现. 这个程序更实用.

程序名称:C09_110.h

```
int NextPaiLie(int n) {
    int i,j,k,t;
    for (i=n-2; i>=0; i--) if(x[i]<x[i+1]) break;
    if (i<0) return(1); // over
    for (j=n-1; j>=i+1; j--) if(x[i]<x[j]) break;
    t=x[i]; x[i]=x[j]; x[j]=t;
    for (j=i+1, k=n-1; j<k; j++, k--) {
        t=x[j]; x[j]=x[k]; x[k]=t;
    }
    return(0); // next step ok!
}
```

程序名称:C09_110.c

#include "stdio.h"

#include "stdlib.h"

int x[100];

#include "C09_060.h"

```
void main() {
    int i,n;
    n=4; // 全排列的元素个数
    for (i=0; i<n; i++) x[i]=i; // 初始化
    for ( ; ; ) {
        for (i=0; i<n; i++) printf("%2d",x[i]+1);
        printf("\n");
        if (NextPaiLie(n)) break;
    }
}
```

§9.10 递归搜索

求一个 9 位数, 该数的每一位均是 1-9 之间的数, 且各数位上的数字互不相同, 使得这个 9 位数从高位开始, 前一位能被 1 整除, 前两位能被 2 整除, 前三位能被 3 整除, …… , 一直到整个 9 位数能被 9 整除. 例: n=381654729

程序名称:C09_120.c

#include "stdio.h"

#include "stdlib.h"

```
int a[10];
void Out(int n) {
    int i,j,s;
    if (n>9) {
        for (i=1; i<=9; i++) printf("%d",a[i]);
        printf("\n"); return;
    }
    for (i=1,s=0; i<n; i++) s=s*10+a[i];
    for (j=1; j<=9; j++) {
        for (i=1; i<n; i++) if (j==a[i]) break;
        if (i>=n) {
            if ((10*s+j)%n==0) {
                a[n]=j; Out(n+1);
            }
        }
    }
}
```

```
void main() {
    Out(1);
}
```

§9.11 作业

1. 用递归实现十进制转换为十六进制

例如: 输入 123 输出 7B

输入 98765 输出 181CD

输入 12345789 输出 75BCD15

2. 求卢卡斯序列

数列: 1, 2, 3, 6, 11, 20, ...

公式: $F(1)=1$

公式: $F(2)=2$

公式: $F(3)=3$

公式: $F(n+3)=F(n+2)+F(n+1)+F(n)$

求 $F(12)$.

3. 利用递归方法求 5!

$5!=1*2*3*4*5$

公式: $F(1)=1, F(n)=n*F(n-1)$

求 $F(5)$.

4. 有 5 个人坐在一起, 问第五个人多少岁? 他说比第 4 个人大 2 岁。问第 4 个人岁数, 他说比第 3 个人大 2 岁。问第三个人, 又说比第 2 人大两岁。问第 2 个人, 说比第一个人大两岁。最后问第一个人, 他说是 10 岁。请问第五个人多大?

5. 给一个不多于 5 位的正整数, 要求: 一、求它是几位数, 二、逆序打印出各位数字。

6. 对策局势的确定

有两堆火柴分别为 7, 10 根, 甲乙二人轮流拿, 要求每次最多拿三根, 最少拿一根, 并且每次只能在其中的一堆中取, 谁最后拿没谁赢, 设甲先拿, 问对甲而言此局势是赢局还是输局?

7. 对策局势的确定

有两堆火柴分别为 7, 10 根, 甲乙二人轮流拿, 每次都有两种取法, 第一种: 要求每次最多拿三根, 最少拿一根, 并且只能在其中的一堆中取, 第二种: 在两堆中同时取, 并且每堆取相同数目, 要求每堆最多拿三根, 最少拿一根, 谁最后拿没谁赢, 设甲先拿, 问对甲而言此局势是赢局还是输局?

8. 对策局势的确定

有三堆火柴分别为 3, 4, 5 根, 甲乙二人轮流拿, 要求每次最少拿一根, 最多可以将整堆全拿

走, 并且每次只能在其中的一堆中取, 谁最后拿没谁赢, 设甲先拿, 问对甲而言此局势是赢局还是输局?

9. 用递归实现图形填充

已知图形, 给定种子点坐标, 要求将图形中所有与种子点连通的所有点都标记为 2. 图形如下:

```
{1, 1, 1, 1, 1, 1, 1, 1, 1},
{1, 0, 0, 0, 1, 0, 0, 0, 1},
{1, 0, 1, 1, 0, 1, 1, 0, 1},
{1, 0, 0, 1, 0, 0, 1, 0, 1},
{1, 1, 0, 1, 1, 0, 0, 0, 1},
{1, 0, 0, 1, 0, 0, 0, 0, 1},
{1, 1, 0, 1, 0, 1, 1, 0, 1},
{1, 0, 0, 0, 1, 0, 0, 0, 1},
{1, 1, 1, 1, 1, 1, 1, 1, 1}
```

第十章 文件操作

§10.1 文件操作的定义

定义文件指针:

```
FILE *fp;
```

打开文件:

```
fp=fopen("文件名称","r"); //读方式打开
```

```
fp=fopen("文件名称","w"); //写方式打开
```

```
fp=fopen("文件名称","rb"); //读方式打开
```

```
fp=fopen("文件名称","wb"); //写方式打开
```

关闭文件:

```
fclose(fp);
```

读文件:

```
fscanf(fp,"%d",&k);
```

```
k=fgetc(fp);
```

写文件

```
fprintf(fp,"%d",k);
```

```
fputc(k,fp);
```

§10.2 写文件

将星型图形写入文本文件中.

```
*  
**  
***  
****  
*****
```

程序名称:C10_010.c

```
#include "stdio.h"
```

```
#include "stdlib.h"
```

```
void main() {  
    int i,j;  
    FILE *fp;  
  
    fp=fopen("星型图形.txt","w");  
    if (fp==NULL) {  
        printf("Error open file!"); exit(1);  
    }  
    for (i=0; i<5; i++) {  
        for (j=0; j<=i; j++) {  
            fprintf(fp,"*");  
        }  
        fprintf(fp,"\n");  
    }  
    fclose(fp);  
}
```

§10.3 数组内容写文件

数据存在数组中, 请将数组中的数据写入文本文件中.

```
13,5,6,3,7,9,2,11,54,33,
```

程序名称:C10_020.c

```
#include "stdio.h"
```

```
#include "stdlib.h"
```

```
void main() {  
    int i,aa[10]={13,5,6,3,7,9,2,11,54,33};  
    FILE *fp;  
  
    fp=fopen("数列1.txt","w");  
    if (fp==NULL) {  
        printf("Error open file!"); exit(1);  
    }  
    for (i=0; i<10; i++) {  
        fprintf(fp,"%d,",aa[i]);  
    }  
    fclose(fp);  
}
```

§10.4 读文件并存于数组中

数据存在文件中, 读入文件中的数据并存于数组中.

```
13,5,6,3,7,9,2,11,54,33,
```

程序名称:C10_030.c

```
#include "stdio.h"
```

```
#include "stdlib.h"
```

```
void main() {  
    int i,aa[10]={0};  
    FILE *fp;  
  
    fp=fopen("数列1.txt","r");  
    if (fp==NULL) {  
        printf("Error open file!"); exit(1);  
    }  
    for (i=0; i<10; i++) {  
        fscanf(fp,"%d",&aa[i]);  
    }  
    fclose(fp);  
    for (i=0; i<10; i++) {  
        printf("%d,",aa[i]);  
    }  
}
```

§10.5 将二维数组内容写入文件中

数据存在于二维数组中, 将数据写入到外部文件中。

```
1,2,3,4,5,
11,12,13,14,15,
21,22,23,24,25,
31,32,33,34,35,
```

程序名称:C10_040.c

```
#include "stdio.h"
```

```
#include "stdlib.h"
```

```
void main() {
    int i,j,aa[5][5]={0};
    FILE *fp;

    fp=fopen("数列2.txt","w");
    if (fp==NULL) {
        printf("Error open file!"); exit(1);
    }
    for (i=0; i<5; i++) { // 生成数据
        for (j=0; j<5; j++) {
            aa[i][j]=10*i+j+1;
        }
    }
    for (i=0; i<5; i++) { // 写入文件
        for (j=0; j<5; j++) {
            fprintf(fp,"%d,",aa[i][j]);
        }
        fprintf(fp,"\n"); //仅用于显示
    }
    fclose(fp);
}
```

§10.6 从文件中读入数据到二维数组中

数据存在于外部文件中, 将数据读入到二维数组中。

```
1,2,3,4,5,
11,12,13,14,15,
21,22,23,24,25,
31,32,33,34,35,
```

程序名称:C10_050.c

```
#include "stdio.h"
```

```
#include "stdlib.h"
```

```
void main() {
    int i,j,aa[5][5]={0};
    FILE *fp;

    fp=fopen("数列2.txt","r");
```

```
if (fp==NULL) {
    printf("Error open file!"); exit(1);
}
for (i=0; i<5; i++) {
    for (j=0; j<5; j++) {
        fscanf(fp,"%d",&aa[i][j]);
    } // 注意,不需要读入回车换行符
}
fclose(fp);
for (i=0; i<5; i++) { // 屏幕显示
    for (j=0; j<5; j++) {
        printf("%d,",aa[i][j]);
    }
    printf("\n");
}
}
```

§10.7 不定长数据的存储

数据存在于二维数组中, 将数据写入到外部文件中。数据共有 4 行, 其中第一个元素表示该行元素个数。

```
5,1,2,3,4,5,
2,12,13,
4,22,23,24,25,
7,32,33,34,35,3,3,9,
```

程序名称:C10_060.c

```
#include "stdio.h"
```

```
#include "stdlib.h"
```

```
void main() {
    int i,j;
    int aa[4][10]={
        {5,1,2,3,4,5},
        {2,12,13},
        {4,22,23,24,25},
        {7,32,33,34,35,3,3,9}
    };
    FILE *fp;

    fp=fopen("数列3.txt","w");
    if (fp==NULL) {
        printf("Error open file!"); exit(1);
    }
    for (i=0; i<4; i++) {
        for (j=0; j<=aa[i][0]; j++) {
            fprintf(fp,"%d,",aa[i][j]);
        }
        fprintf(fp,"\n");
    }
    fclose(fp);
}
```

§10.8 不定长数据的读取

数据存在于外部文件中, 将数据读入到二维数组中.

```
5,1,2,3,4,5,
2,12,13,
4,22,23,24,25,
7,32,33,34,35,3,3,9,
```

程序名称:C10_070.c

```
#include "stdio.h"
```

```
#include "stdlib.h"
```

```
void main() {
    int i,j,aa[4][10]={0};
    FILE *fp;

    fp=fopen("数列3.txt","r");
    if (fp==NULL) {
        printf("Error open file!"); exit(1);
    }
    for (i=0; i<4; i++) {
        fscanf(fp,"%d",&aa[i][0]);
        for (j=1; j<=aa[i][0]; j++) {
            fscanf(fp,"%d",&aa[i][j]);
        }
    }
    fclose(fp);
    for (i=0; i<4; i++) {
        for (j=0; j<=aa[i][0]; j++) {
            printf("%d",aa[i][j]);
        }
        printf("\n");
    }
}
```

§10.9 文件合并

首先以写方式打开两个文件“文本 1.txt”和“文本 2.txt”, 各存放一行文字: “成小事靠才能,” 和 “成大事靠品德.”, 关闭这两个文件, 然后再以读方式打开这两个文件, 并把这两个文件中的信息合并, 输出到一个新文件“文本 3.txt”中.

程序名称:C10_080.c

```
#include "stdio.h"
```

```
#include "stdlib.h"
```

```
void main() {
    char k;
    char sFile1[80]="文本1.txt";
```

```
char sFile2[80]="文本2.txt";
char sFile3[80]="文本3.txt";
FILE *fp1,*fp2,*fp3;

fp1=fopen("文本1.txt","w");
if (fp1==NULL) {
    printf("Error open file!"); exit(1);
}
fprintf(fp1,"成小事靠才能,");
fclose(fp1);

fp2=fopen("文本2.txt","w");
if (fp2==NULL) {
    printf("Error open file!"); exit(1);
}
fprintf(fp2,"成大事靠品德.");
fclose(fp2);

fp1=fopen("文本1.txt","rb");
if (fp1==NULL) {
    printf("Error open file!"); exit(1);
}
fp2=fopen("文本2.txt","rb");
if (fp2==NULL) {
    printf("Error open file!"); exit(1);
}
fp3=fopen("文本3.txt","wb");
if (fp3==NULL) {
    printf("Error open file!"); exit(1);
}

for ( ; ; ) {
    k=fgetc(fp1); if (feof(fp1)) break;
    fputc(k,fp3);
}
for ( ; ; ) {
    k=fgetc(fp2); if (feof(fp2)) break;
    fputc(k,fp3);
}
fclose(fp1);
fclose(fp2);
fclose(fp3);
}
```

§10.10 作业

1. 将星型图形写入文本文件中.

```
*  
***  
*****  
*****  
*****
```

2. 将菲波纳切数列的前 12 项写入到外部文件中.

F1=1, F2=1, F3=2, ...

3. 已知文件中存放了如下数据:

13,5,6,3,7,9,2,11,54,33,

从文件中读入这些数据, 然后计算这些数据的均值. 提示: 均值为 14.3

4. 已知存有不定长数据的文件:

```
5,1,2,3,4,5,  
2,12,13,  
4,22,23,24,25,  
7,32,33,34,35,3,3,9,
```

问, 这个文件中总共存储了多少个有用的数据?
要求不开辟数组空间.

提示: 不在内存中保留文件中的数据, 累计每一行的第一个元素即可, 答案为 18.

5. 某班有 8 名同学, 某次期末考试成绩在文件 “data.txt” 中, 各列含义为: 姓名, 数学, 语文, 外语, 物理, 化学.

```
AA1,80,75,76,88,92  
AA2,58,77,66,88,92  
AA3,70,78,76,78,62  
AA4,60,67,86,68,72  
AA5,55,67,96,58,52  
AA6,90,77,46,48,92  
AA7,98,55,56,88,82  
AA8,83,48,66,88,92
```

试打印下列表格:

- 1) 学期成绩单: 姓名, 总分
- 2) 补考通知单: 姓名, 科目, 分数

第十一章 指针初步

§11.1 指针的定义

定义:

```
int    ab[15],*p1,*p2;
double bc[30],*pp;
char   kk[10],*sp;
```

赋值:

```
p1 = ab;
p1 = &ab[0];
*p1 = 6;
ab[2] = *p1;
p1++;
```

§11.2 利用指针访问数组

由于内存的线性构造, 所以数据全部按照顺序存储.

程序名称:C11_010.c

```
#include "stdio.h"
#include "stdlib.h"
```

```
void main() {
    int i,s,*p;
    int aa[6]={1,3,5,1,6,4};
    p=aa;
    for (i=0,s=0; i<6; i++,p++) {
        s+= *p;
    }
    printf("s=%d\n",s);
}
```

结果为:s=20

§11.3 数组元素倒序

原序 1,3,5,1,6,4

新序 4,6,1,5,3,1

程序名称:C11_020.c

```
#include "stdio.h"
#include "stdlib.h"
```

```
void main() {
    int i,s,*p1,*p2;
    int aa[6]={1,3,5,1,6,4};
    p1=aa;  p2=&aa[5];
    for (; p1<p2; p1++,p2--) {
        s=*p1;  *p1=*p2; *p2=s;
    }
    for (i=0; i<6; i++) printf("%d,",aa[i]);
}
```

结果为:4,6,1,5,3,1,

§11.4 交换两个元素

利用函数交换两个单元中的元素.

程序名称:C11_030.c

```
#include "stdio.h"
#include "stdlib.h"
```

```
void power(int *p1,int *p2) {
    int s;
    s=*p1;  *p1=*p2; *p2=s;
}
```

```
void main() {
    int a,b;

    a=3;  b=4;
    printf("交换前: a=%d,b=%d\n",a,b);
    power(&a,&b);
    printf("交换后: a=%d,b=%d\n",a,b);
}
```

运行结果:

交换前: a=3,b=4

交换后: a=4,b=3

错误的程序:(没能实现交换)

程序名称:C11_040.c

```
#include "stdio.h"
#include "stdlib.h"
```

```
void power(int p1,int p2) {
    int s;
    s=p1;  p1=p2; p2=s;
}
```

```
void main() {
    int a,b;

    a=3;  b=4;
    printf("交换前: a=%d,b=%d\n",a,b);
    power(a,b);
    printf("交换后: a=%d,b=%d\n",a,b);
}
```

运行结果:

交换前: a=3,b=4

交换后: a=3,b=4

指针传递的是地址, 函数通过变更地址对应的数据, 从而实现更改外部程序中的数据.

§11.5 通用排序函数

程序名称:C11_050.c

```
#include "stdio.h"
#include "stdlib.h"
```

```
void Sort(int *ss,int n) {
    int i,j,k;

    for (i=0; i<n; i++) {
        for (j=0; j<n-i-1; j++) {
            if (ss[j]>ss[j+1]) {
                k=ss[j]; ss[j]=ss[j+1]; ss[j+1]=k;
            }
        }
    }
}
```

```
void main() {
    int i,aa[10]={3,5,2,3,6, 1,7,4,6,7};

    Sort(aa,10);
    for (i=0; i<10; i++) printf("%d,",aa[i]);
    printf("\n");
}
```

结果为:1,2,3,3,4,5,6,6,7,7,

§11.6 动态内存开辟

动态开辟 10 个元素空间. 起始位置存在 p 中, 以后可以按照数组的形式进行存储.

程序名称:C11_060.c

```
#include "stdio.h"
#include "stdlib.h"
```

```
void main() {
    int i,*p;

    p=(int*)malloc(sizeof(int)*10);
    if (p==NULL) {
        printf("Error malloc!"); exit(1);
    }
    for (i=0; i<10; i++) p[i]=i;
    for (i=0; i<10; i++) printf("%d,",p[i]);
    printf("\n");
    free(p);
}
```

结果为:0,1,2,3,4,5,6,7,8,9,

§11.7 不定长数据的读写

不定长数据写入外部文件中.

程序名称:C11_070.c

```
#include "stdio.h"
#include "stdlib.h"
```

```
void main() {
    int i,n;
    int aa[10]={32,33,34,35,3,3,9};
    FILE *fp;

    n=7; // 7个有效数据
    fp=fopen("数列4.txt","w");
    if (fp==NULL) {
        printf("Error open file!"); exit(1);
    }
    for(i=0; i<n; i++) fprintf(fp,"%d,",aa[i]);
    fclose(fp);
}
```

开辟内存获取外部不定长数据.

程序名称:C11_080.c

```
#include "stdio.h"
#include "stdlib.h"
```

```
void main() {
    int i,n,*aa;
    FILE *fp;

    fp=fopen("数列4.txt","r");
    if (fp==NULL) {
        printf("Error open file!"); exit(1);
    }
    for (n=0; ; n++) { //i缓存,n计数
        fscanf(fp,"%d",&i); if (feof(fp)) break;
    }
    fseek(fp,0,0); //文件指针归零,从头开始.
    aa=(int*)malloc(sizeof(int)*n); //开辟空间
    if (aa==NULL) { //看看是否开辟成功
        printf("Error malloc!"); exit(1);
    }
    for (i=0; i<n; i++) { //读数据
        fscanf(fp,"%d",&i); //存在&aa[i]中
    }
    fclose(fp);

    printf("%d:",n); //先输出数据的个数
    for (i=0; i<n; i++) printf("%d,",aa[i]);
    free(aa); //释放动态开辟的空间
}

结果为:
7:32,33,34,35,3,3,9,
```


§11.8 利用指针给二维数组赋值

```

0, 1, 2, 3, 4,
5, 6, 7, 8, 9,
10, 11, 12, 13, 14,
15, 16, 17, 18, 19,
程序名称:C11_090.c
#include "stdio.h"
#include "stdlib.h"

void main() {
    int i,j,*p;
    int aa[4][5];

    p=aa[0];
    for (i=0; i<4*5; i++,p++) {
        *p=i;
    }
    p=aa[0]; // 再次赋初值
    for (i=0; i<4; i++) {
        for (j=0; j<5; j++) {
            printf("%3d,",aa[i][j]);
        }
        printf("\n");
    }
}

```

可以使用一维的指针来读取二维甚至多维数组的内容, 本题将 `aa[i][j]` 改为 `p[i*5+j]` 即可。

```
printf("%3d,",p[i*5+j]);
```

§11.9 C++ 上的动态空间开辟

```

程序名称:C11_110.cpp
#include "stdio.h"
#include "stdlib.h"

void main() {
    int i,*p;

    //p=(int*)malloc(sizeof(int)*10);
    p=new int[10];
    if (p==NULL) {
        printf("Error new!"); exit(1);
    }
    for (i=0; i<10; i++) p[i]=i;
    for (i=0; i<10; i++) printf("%d,",p[i]);
    printf("\n");
    delete(p);
}

```

结果为:0,1,2,3,4,5,6,7,8,9,

§11.10 指针与二维数组的关系

```

a 二维数组名,指向一维数组a[0],即0行首地址
&a[0]
-----
&a[1] 1行首地址
a+1
-----
a[0] 0行0列元素地址
*(a+0)
*a
-----
a[1] 1行0列元素a[1][0]的地址
*(a+1)
-----
&a[1][2] 1行2列元素a[1][2]的地址
a[1]+2
*(a+1)+2
-----
a[1][2] 1行2列元素a[1][2]的值
*(a[1]+2)
*(*(a+1)+2)
-----

```

可以将 `*a` 理解成由两部分组成: `a(地址,长度)`

例: `int a[8]` 表示成 `a(地址,4)`

`int a[5][6]` 表示成 `a(地址,4,6)`

`int a[3][7][9]` 表示成 `a(地址,4,9,7)`

`double a[5][6]` 表示成 `a(地址,8,6)`

其中 `int` 的长度是4, `double` 的长度是8.

不难看出`int a[8]`其中的[8]是不参与计算的,同理`int a[5][6]`中的[5]也不参与计算,即第一个维度不参与计算.

设当前地址都是100.

以`int a[8]`为例,表示为`a(地址,4)`,则`a+3`其地址为`100+4*3=112`,`*(a+3)`为`a[3]`的具体值.

以`int a[5][6]`为例,表示为`a(地址,4,6)`,其中`int`的长度是4,`a+1`的地址为`100+1*4*6=124`,形式仍然为`a(新地址,4,6)`,`*(a+1)+3`仍然是地址`a(新地址,4)`,其中新地址=`100+1*4*6+3*4`.

§11.11 作业

1. 已知数组:

1, 3, 5, 1, 6, 4

编写程序, 将上述数组中的元素倒序:

4, 6, 1, 5, 3, 1

要求使用函数实现数据交换, 主程序调用函数完成倒序.

提示: 使用例题中的 power 函数.

2. 编写一个插入排序的函数.

要求主程序将数组以形参传入函数. 提示: 参阅例题.

第十二章 再谈指针

§12.1 定义指针数组

```
int  *ab[15];
int  **p;
int  ***p;
```

§12.2 动态开辟二维数组

```
0,  1,  2,  3,
4,  5,  6,  7,
8,  9, 10, 11,
```

程序名称:C12_010.c

```
#include "stdio.h"
#include "stdlib.h"
```

```
void main() {
    int  i,j,m,n,**p;

    m=3; n=4; // 3行,4列, 开辟空间
    p=(int**)malloc(sizeof(int*)*m);
    if (p==NULL) {
        printf("Error malloc!"); exit(1);
    }
    for (i=0; i<m; i++) p[i]=NULL;
    for (i=0; i<m; i++) {
        p[i]=(int*)malloc(sizeof(int)*n);
        if (p[i]==NULL) {
            printf("Error malloc!"); exit(1);
        }
    }
    // 开辟完成, 向数组赋值
    for (i=0; i<m; i++) {
        for (j=0; j<n; j++) {
            p[i][j]=i*n+j;
        }
    }
    // 输出
    for (i=0; i<m; i++) {
        for (j=0; j<n; j++) {
            printf("%3d,",p[i][j]);
        }
        printf("\n");
    }
    // 释放空间
    for (i=0; i<m; i++) {
        if (p[i]!=NULL) free(p[i]);
    }
    free(p);
}
```

§12.3 获取外部定长数据到内存

数据存在于外部文件中, 首行为矩阵的行和列元素个数, 开辟空间, 然后将矩阵元素读入到内存中.

```
3,4
10, 21, 12, 23,
14, 25, 16, 27,
18, 29, 10, 11,
```

程序名称:C12_020.c

```
#include "stdio.h"
#include "stdlib.h"
```

```
void main() {
    int  i,j,m,n,**p;
    FILE *fp;

    fp=fopen("矩阵1.txt","r");
    if (fp==NULL) {
        printf("Error open file!"); exit(1);
    }
    fscanf(fp,"%d,%d",&m,&n);
    p=(int**)malloc(sizeof(int*)*m);
    if (p==NULL) {
        printf("Error malloc!"); exit(1);
    }
    for (i=0; i<m; i++) p[i]=NULL;
    for (i=0; i<m; i++) {
        p[i]=(int*)malloc(sizeof(int)*n);
        if (p[i]==NULL) {
            printf("Error malloc!"); exit(1);
        }
    }
    // 开辟完成, 读入文件中的数据
    for (i=0; i<m; i++) {
        for (j=0; j<n; j++) {
            fscanf(fp,"%d",&p[i][j]);
        }
    }
    fclose(fp);
    // 输出
    for (i=0; i<m; i++) {
        for (j=0; j<n; j++) {
            printf("%3d,",p[i][j]);
        }
        printf("\n");
    }
    for (i=0; i<m; i++) { // 释放空间
        if (p[i]!=NULL) free(p[i]);
    }
    free(p);
}
```

§12.4 获取外部不定长数据到内存

数据存在于外部文件中, 将数据读入到二维数组中, 不需要的空间不开辟.

```
5,1,2,3,4,5,
2,12,13,
4,22,23,24,25,
7,32,33,34,35,3,3,9,
```

程序名称:C12_030.c

```
#include "stdio.h"
```

```
#include "stdlib.h"
```

```
void main() {
    int i,j,k,m,**p,*q;
    FILE *fp;

    fp=fopen("数列3.txt","r");
    if (fp==NULL) {
        printf("Error open file!"); exit(1);
    }
    for (m=0; ; m++) { // 考察数据行数
        fscanf(fp,"%d",&i); if (feof(fp)) break;
        for (j=1; j<=i; j++) {
            fscanf(fp,"%d",&k); // 跳过数据
        }
    }
    p=(int**)malloc(sizeof(int*)*m);
    if (p==NULL) { // 为每一行建立一个指针
        printf("Error malloc!"); exit(1);
    }
    q=(int*)malloc(sizeof(int)*m); // 存个数
    if (q==NULL) { // 存储每行的元素个数
        printf("Error malloc!"); exit(1);
    }
    fseek(fp,0,0); // 文件指针归零
    for (i=0; i<m; i++) {
        fscanf(fp,"%d",&q[i]); // 读入元素个数
        p[i]=(int*)malloc(sizeof(int)*q[i]);
        if (p[i]==NULL) { // 为每行开辟存储空间
            printf("Error malloc!"); exit(1);
        }
        for (j=0; j<q[i]; j++) { // 获得数据
            fscanf(fp,"%d",&p[i][j]);
        }
    }
    fclose(fp); // 关闭文件
    // 输出
    for (i=0; i<m; i++) {
        for (j=0; j<q[i]; j++) {
            printf("%3d,",p[i][j]);
        }
        printf("\n");
    }
}
```

```
// 释放空间
for (i=0; i<m; i++) {
    if (p[i]!=NULL) free(p[i]);
}
free(p);
free(q);
}
```

§12.5 指针数组的传递

从外部将数据读入到内存中, 调用函数计算全部数据之和.

```
5,1,2,3,4,5,
2,12,13,
4,22,23,24,25,
7,32,33,34,35,3,3,9,
```

程序名称:C12_040.c

```
#include "stdio.h"
```

```
#include "stdlib.h"
```

```
int Sum(int **pp,int *qq,int m) {
    int i,j,ss;
    for (i=0,ss=0; i<m; i++) {
        for (j=0; j<qq[i]; j++) {
            ss+=pp[i][j];
        }
    }
    return(ss);
}
```

```
void main() {
    int i,j,k,m,**p,*q;
    FILE *fp;

    fp=fopen("数列3.txt","r");
    if (fp==NULL) {
        printf("Error open file!"); exit(1);
    }
    for (m=0; ; m++) { // 考察数据行数
        fscanf(fp,"%d",&i); if (feof(fp)) break;
        for (j=1; j<=i; j++) {
            fscanf(fp,"%d",&k); // 跳过数据
        }
    }
    p=(int**)malloc(sizeof(int*)*m);
    if (p==NULL) { // 为每一行建立一个指针
        printf("Error malloc!"); exit(1);
    }
    q=(int*)malloc(sizeof(int)*m); // 存个数
    if (q==NULL) { // 存储每行的元素个数
        printf("Error malloc!"); exit(1);
    }
}
```

```

fseek(fp,0,0); // 文件指针归零
for (i=0; i<m; i++) {
    fscanf(fp,"%d",&q[i]); // 读入元素个数
    p[i]=(int*)malloc(sizeof(int)*q[i]);
    if (p[i]==NULL) { // 为每行开辟存储空间
        printf("Error malloc!"); exit(1);
    }
    for (j=0; j<q[i]; j++) { // 获得数据
        fscanf(fp,"%d",&p[i][j]);
    }
}
fclose(fp); //关闭文件
// 计算全部数据之和,并输出
printf("%d\n",Sum(p,q,m));
// 释放空间
for (i=0; i<m; i++) {
    if (p[i]!=NULL) free(p[i]);
}
free(p);
free(q);
}

```

§12.6 指向指针的指针的指针

某些时候会用到三维数组,若需要动态开辟缓冲区,则需要定义指针完成.

程序名称:C12_050.c

#include "stdio.h"

#include "stdlib.h"

```

void main() {
    int i,j,k,r,m,n,***p;
    int aa[5][3][4];

    // 使用静态数组方式
    for (i=0; i<5; i++) {
        for (j=0; j<3; j++) {
            for (k=0; k<4; k++) {
                aa[i][j][k]=i*3*4+j*4+k;
            }
        }
    }

    // 输出
    for (i=0; i<5; i++) {
        for (j=0; j<3; j++) {
            for (k=0; k<4; k++) {
                printf("%3d",aa[i][j][k]);
            }
            printf("\n");
        }
        printf("\n");
    }
}

```

```

// 使用动态数组方式
r=5; m=3; n=4; //5组,3行,4列,开辟空间
p=(int***)malloc(sizeof(int**)*r);
if (p==NULL) {
    printf("Error malloc!"); exit(1);
}
for (i=0; i<r; i++) p[i]=NULL;
for (i=0; i<r; i++) {
    p[i]=(int**)malloc(sizeof(int*)*m);
    if (p[i]==NULL) {
        printf("Error malloc!"); exit(1);
    }
    for (j=0; j<m; j++) {
        p[i][j]=(int*)malloc(sizeof(int)*n);
        if (p[i][j]==NULL) {
            printf("Error malloc!"); exit(1);
        }
    }
}
// 开辟完成,向数组赋值
for (i=0; i<r; i++) {
    for (j=0; j<m; j++) {
        for (k=0; k<n; k++) {
            p[i][j][k]=i*3*4+j*4+k;
        }
    }
}
// 输出
for (i=0; i<r; i++) {
    for (j=0; j<m; j++) {
        for (k=0; k<n; k++) {
            printf("%3d",p[i][j][k]);
        }
        printf("\n");
    }
    printf("\n");
}
// 释放空间
for (i=0; i<r; i++) {
    for (j=0; j<m; j++) {
        free(p[i][j]);
    }
    free(p[i]);
}
free(p);
}

```

§12.7 作业

1. 数据存在于外部文件中, 将数据读入到二维数组中, 不需要的空间不开辟.

```
1,2,3,4,5,  
12,13,  
22,23,24,25,  
32,33,34,35,3,3,9,
```

计算并输出全部数据之和。读入数据时注意: 每一个数据之后都带有一个逗号, 即使每行数据的最后一个数据也时如此。

说明: 本题较难。由于在外部文件中没有说明数据的行数, 您需要进行一次扫描以便于确定数据的行数, 另外每行数据的个数也没有固定, 所以还需要进行一次扫描以便于确定当前行的元素个数。行与行之间是通过回车换行符来分隔的, 故需要计数回车换行符的个数, 另外有可能出现空行, 需要你的程序跳过空行, 文件最后可能有也可能没有回车换行符, 这些都是需要注意的地方。

第十三章 结构体

§13.1 结构体的定义

日期的结构体定义:

```
struct Date { // 类型定义
    int year; // 年
    int month; // 月
    int day; // 日
};

struct Student { // 类型定义
    int num; // 学号
    int CJ[3]; // 成绩
    int ZCJ; // 总成绩
    struct Date RiQi; // 考试日期
};
```

§13.2 结构体的赋值

程序名称:C13_010.c

```
#include "stdio.h"
```

```
#include "stdlib.h"
```

```
struct Date { // 类型定义
    int year; // 年
    int month; // 月
    int day; // 日
};
```

```
struct Student { // 类型定义
    int num; // 学号
    int CJ[3]; // 成绩
    int ZCJ; // 总成绩
    struct Date RiQi; // 考试日期
};
```

```
void main() {
    int i;
    struct Student XS[2];
```

```
    XS[0].num=101;
    XS[0].CJ[0]=90;
    XS[0].CJ[1]=80;
    XS[0].CJ[2]=80;
    XS[0].RiQi.year=2013;
    XS[0].RiQi.month=3;
    XS[0].RiQi.day=4;
    XS[0].ZCJ=0;
    for (i=0; i<3; i++) {
        XS[0].ZCJ += XS[0].CJ[i];
    }
```

```
    XS[1].num=102;
    XS[1].CJ[0]=80;
    XS[1].CJ[1]=80;
    XS[1].CJ[2]=70;
    XS[1].RiQi.year=2013;
    XS[1].RiQi.month=3;
    XS[1].RiQi.day=4;
    XS[1].ZCJ=0;
    for (i=0; i<3; i++) {
        XS[1].ZCJ += XS[1].CJ[i];
    }

    printf("学号 数学 语文 外语");
    printf(" 总成绩 考试日期\n");
    for (i=0; i<2; i++) {
        printf(" %d %d %d %d",XS[i].num,
            XS[i].CJ[0],XS[i].CJ[1],XS[i].CJ[2]);
        printf(" %d %d-%d-%d\n",
            XS[i].ZCJ,XS[i].RiQi.year,
            XS[i].RiQi.month,XS[i].RiQi.day);
    }
}
```

结果如下:

学号	数学	语文	外语	总成绩	考试日期
101	90	80	80	250	2013-3-4
102	80	80	70	230	2013-3-4

§13.3 结构体的指针

程序名称:C13_020.c

```
#include "stdio.h"
```

```
#include "stdlib.h"
```

```
struct Date { // 类型定义
    int year; // 年
    int month; // 月
    int day; // 日
};
```

```
struct Student { // 类型定义
    int num; // 学号
    int CJ[3]; // 成绩
    int ZCJ; // 总成绩
    struct Date RiQi; // 考试日期
};
```

```
void main() {
    int i;
    struct Student XS[2],*p;
```

```

p=&XS[0];
p->num=101;
p->CJ[0]=90;
p->CJ[1]=80;
p->CJ[2]=80;
p->RiQi.year=2013;
p->RiQi.month=3;
p->RiQi.day=4;
p->ZCJ=0;
for (i=0; i<3; i++) {
    p->ZCJ += p->CJ[i];
}

p=&XS[1];
p->num=102;
p->CJ[0]=80;
p->CJ[1]=80;
p->CJ[2]=70;
p->RiQi.year=2013;
p->RiQi.month=3;
p->RiQi.day=4;
p->ZCJ=0;
for (i=0; i<3; i++) {
    p->ZCJ += p->CJ[i];
}

printf("学号 数学 语文 外语");
printf(" 总成绩  考试日期\n");
for (i=0; i<2; i++) {
    printf(" %d %d %d %d",XS[i].num,
        XS[i].CJ[0],XS[i].CJ[1],XS[i].CJ[2]);
    printf(" %d %d-%d-%d\n",
        XS[i].ZCJ,XS[i].RiQi.year,
        XS[i].RiQi.month,XS[i].RiQi.day);
}
}

```

§13.4 简单链表

程序名称:C13_030.c

```
#include "stdio.h"
```

```
#include "stdlib.h"
```

```

struct Student { // 类型定义
    int num;      // 学号
    int CJ[3];    // 成绩
    int ZCJ;      // 总成绩
    struct Student *next; // 指针
};

```

```
typedef struct Student STUDENT;
```

```
void main() {
```

```

int i;
STUDENT *head,*p;

p=(STUDENT *)malloc(sizeof(STUDENT));
p->num=101;
p->CJ[0]=90;
p->CJ[1]=80;
p->CJ[2]=80;
p->ZCJ=0;
p->next=NULL;
for (i=0; i<3; i++) {
    p->ZCJ += p->CJ[i];
}
head=p;

p=(STUDENT *)malloc(sizeof(STUDENT));
p->num=102;
p->CJ[0]=80;
p->CJ[1]=80;
p->CJ[2]=70;
p->next=NULL;
p->ZCJ=0;
for (i=0; i<3; i++) {
    p->ZCJ += p->CJ[i];
}
head->next=p;

printf("学号 数学 语文 外语 总成绩\n");
p=head;
for (i=0; i<2; i++) {
    printf(" %d %d %d %d",p->num,
        p->CJ[0],p->CJ[1],p->CJ[2]);
    printf(" %d\n",p->ZCJ);
    p=p->next;
}
// 释放空间
p=head; head=head->next; free(p);
p=head; head=head->next; free(p);
}

```

§13.5 链表应用

数据存在外部文件中,第一列为序号,其余三列为具体成绩.

```

101,90,80,80
102,80,80,70
103,70,80,60
104,80,80,90
105,60,80,70

```

程序名称:C13_040.c

```
#include "stdio.h"
```

```
#include "stdlib.h"
```



```

struct Student { // 类型定义
    int  num;      // 学号
    int  CJ[3];    // 成绩
    int  ZCJ;      // 总成绩
    struct Student *next; // 指针
};

typedef struct Student STUDENT;

void main() {
    int  i,n,nn,n1,n2,n3;
    STUDENT *head,*tai,*p;
    FILE  *fp;

    fp=fopen("学生成绩.txt","r");
    if (fp==NULL) {
        printf("Error open file!"); exit(1);
    }
    for (n=0; ; n++) {
        fscanf(fp,"%d,%d,%d,%d",&nn,&n1,&n2,&n3);
        if (feof(fp)) break;
        p=(STUDENT *)malloc(sizeof(STUDENT));
        p->num=nn;
        p->CJ[0]=n1;
        p->CJ[1]=n2;
        p->CJ[2]=n3;
        p->next=NULL;
        p->ZCJ=0;
        for (i=0; i<3; i++) {
            p->ZCJ += p->CJ[i];
        }
        if (n==0) { head=p; tai=p; }
        else { tai->next=p; tai=p; }
    }
    fclose(fp);

    printf("学号 数学 语文 外语 总成绩\n");
    for (p=head; p!=NULL; p=p->next) {
        printf(" %d  %d  %d  %d",p->num,
            p->CJ[0],p->CJ[1],p->CJ[2]);
        printf(" %d\n",p->ZCJ);
    }
    // 释放空间
    for (p=head; p!=NULL; p=p->next) {
        head=head->next; free(p);
    }
}

```

运行结果:

学号	数学	语文	外语	总成绩
101	90	80	80	250
102	80	80	70	230
103	70	80	60	210

104	80	80	90	250
105	60	80	70	210

§13.6 C++ 的类定义

程序名称:C13_050.cpp

```
#include "stdio.h"
```

```
#include "stdlib.h"
```

```

class Student { // 类型定义
public:
    int  num;      // 学号
    int  CJ[3];    // 成绩
    int  ZCJ;      // 总成绩
public:
    void display(); // 显示
    void sum();     // 计算总成绩
};

void Student::display() { // 显示
    printf(" %d  %d  %d  %d  %d\n",
        num,CJ[0],CJ[1],CJ[2],ZCJ);
}

void Student::sum() { // 计算总成绩
    int  i;
    for (i=0,ZCJ=0; i<3; i++) ZCJ+=CJ[i];
}

void main() {
    Student XS[2];

    printf("学号 数学 语文 外语 总成绩\n");

    XS[0].num=101;
    XS[0].CJ[0]=90;
    XS[0].CJ[1]=80;
    XS[0].CJ[2]=80;
    XS[0].sum();
    XS[0].display();

    XS[1].num=102;
    XS[1].CJ[0]=80;
    XS[1].CJ[1]=80;
    XS[1].CJ[2]=70;
    XS[1].sum();
    XS[1].display();
}

```

§13.7 作业

1. 数据存在外部文件中, 第一列为序号, 其余三列为具体成绩.

101,90,80,80

102,80,80,70

103,70,80,60

104,80,80,90

105,60,80,70

使用结构体存储并输出每个人的成绩、总成绩以及平均成绩. 注意, 最后一行需要有回车换行符.