**题目号 题目名**

**实验1-1 Hello World!**

**实验1-2 Welcome to You!**

**实验1-3 Programming in C is fun!**

**实验1-4 输出三角形**

**实验1-5 输出菱形图案**

**实验1-6 输出带框文字**

**实验1-7 What is a computer?**

**实验1-8 输出倒三角图案**

**实验2-1-1 计算摄氏温度**

**实验2-1-2 温度转换**

**实验2-1-3 计算物体自由下落的距离**

**实验2-1-4 计算平均分**

**实验2-1-5 将x的平方赋值给y**

**实验2-1-6 计算华氏温度**

**实验2-1-7 整数152的各位数字**

**实验2-2-1 计算分段函数[1]**

**实验2-2-2 计算摄氏温度**

**实验2-2-3 计算存款利息**

**实验2-2-4 计算分段函数[2]**

**实验2-2-5 求整数均值**

**实验2-2-6 计算分段函数[3]**

**实验2-2-7 整数四则运算**

**实验2-2-8 阶梯电价**

**实验2-2-9 计算火车运行时间**

**实验2-3-1 求1到100的和**

**实验2-3-2 求N分之一序列前N项和**

**实验2-3-3 求奇数分之一序列前N项和**

**实验2-3-4 求简单交错序列前N项和**

**实验2-3-5 输出华氏-摄氏温度转换表**

**实验2-3-6 求交错序列前N项和**

**实验2-3-7 求平方与倒数序列的部分和**

**实验2-4-2 生成3的乘方表**

**实验2-4-3 求平方根序列前N项和**

**实验2-4-4 求阶乘序列前N项和**

**实验2-4-6 求幂之和**

**实验2-4-7 求组合数**

**实验3-1 求一元二次方程的根**

**实验3-2 计算符号函数的值**

**实验3-3 比较大小**

**实验3-4 统计字符**

**实验3-5 查询水果价格**

**实验3-6 计算个人所得税**

**实验3-7 统计学生成绩**

**实验3-8 输出三角形面积和周长**

**实验3-9 三天打鱼两天晒网**

**实验3-10 高速公路超速处罚**

**实验3-11 计算油费**

**实验4-1-1 最大公约数和最小公倍数**

**实验4-1-2 求奇数和**

**实验4-1-3 找出最小值**

**实验4-1-4 求整数的位数及各位数字之和**

**实验4-1-5 韩信点兵**

**实验4-1-6 求分数序列前N项和**

**实验4-1-7 特殊a串数列求和**

**实验4-1-8 求给定精度的简单交错序列部分和**

**实验4-1-9 猜数字游戏**

**实验4-1-10 兔子繁衍问题**

**实验4-1-11 高空坠球**

**实验4-1-12 黑洞数**

**实验4-2-2 求e的近似值**

**实验4-2-3 验证“哥德巴赫猜想”**

**实验4-2-4 换硬币**

**实验4-2-5 水仙花数**

**实验4-2-6 输出三角形字符阵列**

**实验4-2-7 找完数**

**实验4-2-8 输出整数各位数字**

**实验4-2-9 梅森数**

**实验6-1 近似求PI**

**实验6-8 简单计算器**

**实验6-9 统计一行文本的单词个数**

**实验7-1-1 简化的插入排序**

**实验7-1-2 求最大值及其下标**

**实验7-1-3 将数组中的数逆序存放**

**实验7-1-4 找出不是两个数组共有的元素**

**实验7-1-5 选择法排序**

**实验7-1-6 求一批整数中出现最多的个位数字**

**实验7-1-7 查找整数**

**实验7-1-8 输出数组元素**

**实验7-1-9 数字加密**

**实验7-1-10 交换最小值和最大值**

**实验7-1-11 求整数序列中出现次数最多的数**

**实验7-1-12 组个最小数**

**实验7-1-13 装箱问题**

**实验7-2-1 求矩阵各行元素之和**

**实验7-2-2 矩阵运算**

**实验7-2-3 求矩阵的局部极大值**

**实验7-2-4 计算天数**

**实验7-2-5 判断上三角矩阵**

**实验7-2-6 打印杨辉三角**

**实验7-2-7 方阵循环右移**

**实验7-2-8 找鞍点**

**实验7-2-9 螺旋方阵**

**实验7-2-10 简易连连看**

**实验7-3-1 字符串逆序**

**实验7-3-2 查找指定字符**

**实验7-3-3 统计大写辅音字母**

**实验7-3-4 字符串替换**

**实验7-3-5 输出大写英文字母**

**实验7-3-6 字符串转换成十进制整数**

**实验7-3-7 字符转换**

**实验7-3-8 统计字符出现次数**

**实验7-3-9 字符串字母大小写转换**

**实验7-3-10 删除重复字符**

**实验8-1-9 输出学生成绩**

**实验8-2-1 找最小的字符串**

**实验8-2-2 找最长的字符串**

**实验8-2-8 字符串排序**

**实验8-2-10 IP地址转换**

**实验9-1 计算职工工资**

**实验9-2 时间换算**

**实验9-3 计算平均成绩**

**实验9-5 查找书籍**

**实验9-7 找出总分最高的学生**

**实验9-8 通讯录排序**

**实验9-9 有理数比较**

**实验9-10 平面向量加法**

**实验10-1 圆形体体积计算器**

**实验11-1-1 英文单词排序**

**实验11-1-7 藏头诗**

**实验11-1-9 藏尾诗**

**函数题：**

**题目号 题目名**

**实验2-4-1 统计各位数字之和是5的数**

**实验2-4-5 简单实现x的n次方**

**实验5-1 使用函数计算两个复数之积**

**实验5-2 符号函数**

**实验5-3 使用函数求奇数和**

**实验5-4 使用函数计算两点间的距离**

**实验5-5 使用函数求素数和**

**实验5-6 使用函数判断完全平方数**

**实验5-7 使用函数求1到10的阶乘和**

**实验5-8 使用函数统计指定数字的个数**

**实验5-9 使用函数输出水仙花数**

**实验5-10 使用函数求余弦函数的近似值**

**实验5-11 使用函数求最大公约数**

**实验6-2 分类统计字符个数**

**实验6-3 使用函数求特殊a串数列和**

**实验6-4 使用函数输出指定范围内的完数**

**实验6-5 使用函数输出指定范围内的Fibonacci数**

**实验6-6 使用函数验证哥德巴赫猜想**

**实验6-7 使用函数输出一个整数的逆序数**

**实验8-1-1 利用指针找最大值**

**实验8-1-2 计算两数的和与差**

**实验8-1-3 拆分实数的整数与小数部分**

**实验8-1-4 使用函数的选择法排序**

**实验8-1-5 在数组中查找指定元素**

**实验8-1-6 函数实现字符串逆序**

**实验8-1-7 数组循环右移**

**实验8-1-8 报数**

**实验8-2-3 删除字符**

**实验8-2-4 使用函数实现字符串部分复制**

**实验8-2-5 判断回文字符串**

**实验8-2-6 分类统计各类字符个数**

**实验8-2-7 字符串的连接**

**实验8-2-9 长整数转化成16进制字符串**

**实验9-4 计算两个复数之积**

**实验9-6 按等级统计学生成绩**

**实验10-2 判断满足条件的三位数**

**实验10-3 递归求阶乘和**

**实验10-4 递归实现指数函数**

**实验10-5 递归求简单交错幂级数的部分和**

**实验10-6 递归计算Ackermenn函数**

**实验10-7 递归求Fabonacci数列**

**实验10-8 递归计算P函数**

**实验10-9 十进制转换二进制**

**实验10-10 递归实现顺序输出整数**

**实验11-1-2 输出月份英文名**

**实验11-1-3 查找星期**

**实验11-1-4 计算最长的字符串长度**

**实验11-1-6 指定位置输出字符串**

**实验11-1-8 查找子串**

**实验11-2-1 建立学生信息链表**

**实验11-2-2 学生成绩链表处理**

**实验11-2-3 逆序数据建立链表**

**实验11-2-4 删除单链表偶数节点**

**实验11-2-5 链表拼接**

**实验11-2-6 奇数值结点链表**

**实验11-2-7 统计专业人数**

**实验11-2-8 单链表结点删除**

**实验11-2-9 链表逆置**

**编程题**

**实验1-1 Hello World! (5 分)**

#include <stdio.h>

int main()

{

printf("Hello World!");

return 0;

}

**实验1-2 Welcome to You! (5 分)**

#include<stdio.h>

int main()

{

printf("Welcome to You!");

return 0;

}

**实验1-3 Programming in C is fun! (5 分)**

#include <stdio.h>

int main()

{

printf("Programming in C is fun!");

return 0;

}

**实验1-4 输出三角形 (5 分)**

#include <stdio.h>

int main()

{

int i,n=4;

printf("\*\*\*\*\n");

printf("\*\*\*\n");

printf("\*\*\n");

printf("\*\n");

return 0;

}

**实验1-5 输出菱形图案 (5 分)**

#include<stdio.h>

int main()

{

printf(" A\n");

printf("A A\n");

printf(" A\n");

return 0;

}

**实验1-6 输出带框文字 (5 分)**

#include <stdio.h>

int main()

{

printf("\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf(" Welcome\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\n");

return 0;

}

**实验1-7 What is a computer? (5 分)**

#include<stdio.h>

int main()

{

printf("What is a computer?");

return 0;

}

**实验1-8 输出倒三角图案 (5 分)**

#include <stdio.h>

int main()

{

printf("\* \* \* \*\n");

printf(" \* \* \*\n");

printf(" \* \*\n");

printf(" \*\n");

return 0;

}

**实验2-1-1 计算摄氏温度 (5 分)**

#include <stdio.h>

int main()

{

printf("fahr = 100, celsius = %d",5\*(100-32)/9);

return 0;

}

**实验2-1-2 温度转换 (5 分)**

#include <stdio.h>

int main()

{

printf("fahr = 150, celsius = %d",5\*(150-32)/9);

return 0;

}

**实验2-1-3 计算物体自由下落的距离 (5 分)**

#include <stdio.h>

int main()

{

printf("height = 45.00");

return 0;

}

**实验2-1-4 计算平均分 (5 分)**

#include<stdio.h>

int main(void)

{

printf("math = 87, eng = 72, comp = 93, average = %d", (87 + 72 + 93) / 3);

return 0;

}

**实验2-1-5 将x的平方赋值给y (5 分)**

#include <stdio.h>

int main()

{

printf("9 = 3 \* 3\n");

printf("3 \* 3 = 9");

return 0;

}

**实验2-1-6 计算华氏温度 (5 分)**

#include <stdio.h>

int main()

{

printf("celsius = 26, fahr = %d",9\*26/5+32);

return 0;

}

**实验2-1-7 整数152的各位数字 (10 分)**

#include <stdio.h>

int main()

{

printf("152 = 2 + 5\*10 + 1\*100");

return 0;

}

**实验2-2-1 计算分段函数[1] (10 分)**

#include <stdio.h>

int main()

{

float x,y;

scanf("%f",&x);

if(x!=0)

y=1/x;

else

y=0.0;

printf("f(%.1f) = %.1f",x,y);

return 0;

}

**实验2-2-2 计算摄氏温度 (10 分)**

#include <stdio.h>

int main()

{

int f;

scanf("%d",&f);

printf("Celsius = %d",5\*(f-32)/9);

return 0;

}

**实验2-2-3 计算存款利息 (10 分)**

#include <stdio.h>

#include <math.h>

int main()

{

float money, year, rate;

scanf("%f %f %f",&money, &year, &rate);

printf("interest = %.2f",money\*pow((1+rate),year)-money);

return 0;

}

**实验2-2-4 计算分段函数[2] (10 分)**

#include <stdio.h>

#include <math.h>

int main()

{

float x,result;

scanf("%f",&x);

if(x>=0)

result=sqrt(x);

else

result=pow(x+1,2)+2\*x+1/x;

printf("f(%.2f) = %.2f",x,result);

return 0;

}

**实验2-2-5 求整数均值 (10 分)**

#include <stdio.h>

int main()

{

int sum = 0, a[4];

for(int i = 0; i < 4;i++)

{

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

sum += a[i];

}

printf("Sum = %d; Average = %.1f",sum,sum/4.0);

return 0;

}

**实验2-2-6 计算分段函数[3] (10 分)**

#include <stdio.h>

int main()

{

float x, y;

scanf("%f",&x);

if(x != 10)

y = x;

else

y = 1.0/x;

printf("f(%.1f) = %.1f",x, y);

return 0;

}

**实验2-2-7 整数四则运算 (10 分)**

#include <stdio.h>

int main()

{

int a, b;

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

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

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

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

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

return 0;

}

**实验2-2-8 阶梯电价 (15 分)**

#include <stdio.h>

int main()

{

int ele;

float cost;

scanf("%d",&ele);

if(ele <= 50)

cost = ele\*0.53;

else

cost = 50\*0.53+(ele - 50)\*(0.53 + 0.05);

if(ele >= 0)

printf("cost = %.2f",cost);

else

printf("Invalid Value!");

return 0;

}

**实验2-2-9 计算火车运行时间 (15 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

int main()

{

int str, end, str\_h, str\_m, end\_h, end\_m;

scanf("%d %d", &str, &end);

str\_h = str / 100; str\_m = str % 100;

end\_h = end / 100; end\_m = end % 100;

if (end\_m - str\_m < 0)

printf("%02d:%02d", end\_h - str\_h - 1, 60 - str\_m + end\_m);

else

printf("%02d:%02d", end\_h - str\_h, end\_m - str\_m);

return 0;

}

**实验2-3-1 求1到100的和 (10 分)**

#include <stdio.h>

int main()

{

int sum = 0;

for(int i = 1;i <= 100;i++)

sum += i;

printf("sum = %d",sum);

return 0;

}

**实验2-3-2 求N分之一序列前N项和 (15 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

int main()

{

int n;

double s = 0;

scanf("%d", &n);

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

s += 1.0 / i;

printf("sum = %.6lf", s);

return 0;

}

**实验2-3-3 求奇数分之一序列前N项和 (15 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

int main()

{

int n;

double sum = 0, t = 1;

scanf("%d", &n);

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

{

sum += 1.0 / t;

t += 2;

}

printf("sum = %.6lf", sum);

return 0;

}

**实验2-3-4 求简单交错序列前N项和 (15 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

int main()

{

int n;

double sum = 0, x = 1;

scanf("%d", &n);

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

{

if (i % 2 == 0)

sum += 1.0 / x;

else

sum -= 1.0 / x;

x += 3;

}

printf("sum = %.3lf", sum);

return 0;

}

**实验2-3-5 输出华氏-摄氏温度转换表 (15 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

#include <stdlib.h>

int main()

{

int lower, upper;

scanf("%d %d", &lower, &upper);

if (lower > upper || upper > 100)

{

printf("Invalid.");

exit(0);

}

printf("fahr celsius\n");

do

{

printf("%d%6.1f\n", lower, 5.0 \* (lower - 32) / 9);

lower += 2;

} while (lower <= upper);

return 0;

}

**实验2-3-6 求交错序列前N项和 (15 分)**

#include <stdio.h>

int main()

{

int n;

float fenz, fenm, sum = 0;

fenz = fenm = 1;

scanf("%d", &n);

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

{

if (i % 2 == 0)

sum += fenz / fenm;

else

sum -= fenz / fenm;

fenz++;

fenm += 2;

}

printf("%.3f", sum);

return 0;

}

**实验2-3-7 求平方与倒数序列的部分和 (15 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

int main()

{

int m, n, i = 0;

double sum = 0;

scanf("%d %d", &m, &n);

while (m <= n)

{

if (i % 2 == 0)

sum += m \* m;

else

{

sum += 1.0 / m;

m++;

}

i++;

}

printf("sum = %.6lf", sum);

return 0;

}

**实验2-4-2 生成3的乘方表 (15 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

#include <math.h>

int main()

{

int n;

scanf("%d", &n);

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

printf("pow(3,%d) = %.0f\n", i, pow(3, i));

return 0;

}

**实验2-4-3 求平方根序列前N项和 (15 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

#include<math.h>

int main()

{

int n;

double sum = 0;

scanf("%d", &n);

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

sum += sqrt(i);

printf("sum = %.2f", sum);

return 0;

}

**实验2-4-4 求阶乘序列前N项和 (15 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

int fac(int n);

int main()

{

int n, sum = 0;

scanf("%d", &n);

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

sum += fac(i);

printf("%d", sum);

return 0;

}

int fac(int n)

{

int s = 1;

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

s \*= i;

return s;

}

**实验2-4-6 求幂之和 (15 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

#include <math.h>

int main(void)

{

int n, sum = 0;

scanf("%d", &n);

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

sum += pow(2, i);

printf("result = %d", sum);

return 0;

}

**实验2-4-7 求组合数 (15 分)**

#include <stdio.h>

double fact(int n);

int main()

{

int m,n;

double sum;

scanf("%d %d",&m,&n);

sum=fact(n)/(fact(m)\*fact(n-m));

printf("result = %.0lf",sum);

return 0;

}

double fact(int x)

{

double s=1;

int i;

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

s\*=i;

return s;

}

**实验3-1 求一元二次方程的根 (20 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

#include <math.h>

int main(void)

{

float a, b, c, de;

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

de = b \* b - 4 \* a \* c;

if (a == 0 && b == 0)

{

if (c != 0)

printf("Not An Equation");

else

printf("Zero Equation");

}

else if (a == 0 && b != 0)

{

if (c != 0)

printf("%.2f", -c / b);

else

printf("%.2f", 0);

}

else if (de > 0)

printf("%.2f\n%.2f", (-b + sqrt(de)) / (2 \* a), (-b - sqrt(de)) / (2 \* a));

else if (de == 0)

printf("%.2f", -b / (2 \* a));

else

{

if (-b / (2 \* a) != 0)

{

printf("%.2f+%.2fi\n", -b / (2 \* a), sqrt(-de) / (2 \* a));

printf("%.2f%.2fi\n", -b / (2 \* a), -sqrt(-de) / (2 \* a));

}

else

{

printf("0.00+%.2fi\n", sqrt(-de) / (2 \* a));

printf("0.00%.2fi\n", -sqrt(-de) / (2 \* a));

}

}

return 0;

}

**实验3-2 计算符号函数的值 (10 分)**

#include <stdio.h>

#include <math.h>

int main()

{

int n, y;

scanf("%d",&n);

if(n < 0)

y = -1;

else if(n == 0)

y = 0;

else

y = 1;

printf("sign(%d) = %d", n, y);

return 0;

}

**实验3-3 比较大小 (10 分)**

#include <stdio.h>

int main()

{

int a[3], temp;

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

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

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

{

for(int j = i; j < 3; j++)

{

if(a[i] > a[j])

{

temp = a[j];

a[j] = a[i];

a[i] = temp;

}

}

}

printf("%d->%d->%d",a[0], a[1], a[2]);

return 0;

}

**实验3-4 统计字符 (15 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

int main()

{

int letter, blank, digit, other;

char a[10];

letter = blank = digit = other = 0;

for (int i = 0; i < 10; i++)

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

for (int i = 0; i < 10; i++)

{

if (a[i] >= 'a' && a[i] <= 'z' || a[i] >= 'A' && a[i] <='Z')

letter++;

else if (a[i] == ' ' || a[i] == '\n')

blank++;

else if (a[i] >= '0' && a[i] <= '9')

digit++;

else

other++;

}

printf("letter = %d, blank = %d, digit = %d, other = %d", letter, blank, digit, other);

return 0;

}

**实验3-5 查询水果价格 (15 分)**

#include <stdio.h>

#include <stdlib.h>

int main()

{

int i,choice;

printf("[1] apple\n[2] pear\n[3] orange\n[4] grape\n[0] exit\n");

for(i = 0;i < 5;i++)

{

scanf("%d",&choice);

switch(choice)

{

case 1:printf("price = 3.00\n");

break;

case 2:printf("price = 2.50\n");

break;

case 3:printf("price = 4.10\n");

break;

case 4:printf("price = 10.20\n");

break;

case 0:

exit(0);

default:printf("price = 0.00\n");

break;

}

}

return 0;

}

**实验3-6 计算个人所得税 (10 分)**

#include <stdio.h>

int main()

{

float s, rate, salary;

scanf("%f",&salary);

if(salary <= 1600)

s = 0;

else if(salary <= 2500)

s = 0.05\*(salary - 1600);

else if(salary <= 3500)

s = 0.1\*(salary - 1600);

else if(salary <= 4500)

s = 0.15\*(salary - 1600);

else

s = 0.2\*(salary - 1600);

printf("%.2f",s);

return 0;

}

**实验3-7 统计学生成绩 (15 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

int main()

{

int n, a, b, c, d, e, score;

a = b = c = d = e = 0;

scanf("%d", &n);

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

{

scanf("%d", &score);

if (score >= 90)

a++;

else if (score >= 80)

b++;

else if (score >= 70)

c++;

else if (score >= 60)

d++;

else

e++;

}

printf("%d %d %d %d %d", a, b, c, d, e);

return 0;

}

**实验3-8 输出三角形面积和周长 (15 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

#include <math.h>

int main()

{

float a, b, c, s;

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

s = (a + b + c) / 2;

if (a + b > c && a + c > b && b + c > a)

printf("area = %.2f; perimeter = %.2f", sqrt(s \* (s - a) \* (s - b) \* (s - c)), 2 \* s);

else

printf("These sides do not correspond to a valid triangle");

return 0;

}

**实验3-9 三天打鱼两天晒网 (15 分)**

#include <stdio.h>

int main()

{

int n,x;

scanf("%d",&n);

if(n <= 5){

if(x - 3 <= 0)

printf("Fishing in day %d",n);

else

printf("Drying in day %d",n);

}

else{

x = n%5;

if(x == 0)

printf("Drying in day %d",n);

else if(x - 3 <= 0)

printf("Fishing in day %d",n);

else

printf("Drying in day %d",n);

}

return 0;

}

**实验3-10 高速公路超速处罚 (15 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

int main()

{

int v, limv;

scanf("%d %d", &v, &limv);

if ((float)v / limv - 1 < 0.1)

printf("OK");

else if ((float)v / limv - 1 < 0.5)

printf("Exceed %.0f%%. Ticket 200", ((float)v / limv - 1) \* 100);

else

printf("Exceed %.0f%%. License Revoked", ((float)v / limv - 1) \* 100);

return 0;

}

**实验3-11 计算油费 (15 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

int main()

{

int a, b;

char c;

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

if (b == 90)

{

if (c == 'm')

printf("%.2f", a \* 6.95 \* 0.95);

if(c=='e')

printf("%.2f", a \* 6.95 \* 0.97);

}

if (b == 93)

{

if (c == 'm')

printf("%.2f", a \* 7.44 \* 0.95);

if (c == 'e')

printf("%.2f", a \* 7.44 \* 0.97);

}

if (b == 97)

{

if (c == 'm')

printf("%.2f", a \* 7.93 \* 0.95);

if (c == 'e')

printf("%.2f", a \* 7.93 \* 0.97);

}

return 0;

}

**实验4-1-1 最大公约数和最小公倍数 (15 分)**

#include<stdio.h>

int main()

{

int i,a,b,r,d;

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

r=a\*b;

if(a<b) {

i=a;

a=b;

b=i;

}

do

{

i=a%b;

a=b;

b=i;

}while(i!=0);

d=r/a;

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

return 0;

}

**实验4-1-2 求奇数和 (15 分)**

#include <stdio.h>

int main()

{

int n,sum;

scanf("%d",&n);

while(n>0){

if(n%2!=0)

sum += n;

scanf("%d",&n);

}

printf("%d",sum);

return 0;

}

**实验4-1-3 找出最小值 (20 分)**

#include <stdio.h>

int main()

{

int i,m,n,min;

scanf("%d",&n);

scanf("%d",&m);

min=m;

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

scanf("%d",&m);

if(m<min)

min=m;

}

printf("min = %d",min);

return 0;

}

**实验4-1-4 求整数的位数及各位数字之和 (15 分)**

#include <stdio.h>

int main()

{

int n,w=1,sum=0;

scanf("%d",&n);

while(n/10>0){

sum += n%10;

w++;

n=n/10;

}

printf("%d %d",w,sum+n);

return 0;

}

**实验4-1-5 韩信点兵 (10 分)**

#include <stdio.h>

int main()

{

int i;

for(i=1; ;i++){

if((i-1)%5==0)

if((i-5)%6==0)

if((i-4)%7==0)

if((i-10)%11==0)

break;

}

printf("%d",i);

return 0;

}

**实验4-1-6 求分数序列前N项和 (15 分)**

#include <stdio.h>

int main()

{

int n,i;

double a = 2, b = 1,r,sum = 0;

r = b;

scanf("%d", &n);

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

sum += a / b;

b = a;

a = a + r;

r = b;

}

printf("%.2f",sum);

return 0;

}

**实验4-1-7 特殊a串数列求和 (20 分)**

#include <stdio.h>

int main()

{

int i,a,r,n,s = 0;

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

r = a;

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

s += a;

a = (a\*10)+r;

}

printf("s = %d",s);

return 0;

}

**实验4-1-8 求给定精度的简单交错序列部分和 (15 分)**

#include<stdio.h>

#include<math.h>

int main()

{

int i = 1, flag = 1;

double tmp, eps, sum = 0;

scanf( "%lf", &eps );

do

{

tmp = flag \* 1.0 / ( 3 \* i - 2 );

i++;

sum+= tmp;

flag = -flag;

}while( fabs( tmp ) > eps );

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

return 0;

}

**实验4-1-9 猜数字游戏 (15 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

#include <stdlib.h>

int main()

{

int s, n, t = 0, a; //s为随机数，n为猜测最大次数 ,t为猜的次数 ,a为用户每次输入的数

scanf("%d %d", &s, &n);

for (int i = 1; ; i++)

{

scanf("%d", &a);

t++;

if (t > n)

{

printf("Game Over\n");

exit(0);

}

if (a < 0)

{

printf("Game Over\n");

exit(0);

}

if (a > s)

printf("Too big\n");

else if (a < s)

printf("Too small\n");

else

break;

}

if (t == 1)

printf("Bingo!\n");

else if (t > 1 && t <= 3)

printf("Lucky You!\n");

else if (t > 3 && t <= n)

printf("Good Guess!\n");

return 0;

}

**实验4-1-10 兔子繁衍问题 (15 分)**

#include <stdio.h>

int main()

{

int n,a = 1,b = 1,c,month = 2,i;

scanf("%d",&n);

if(n==1)

printf("1");

else{

for(i = 1; ;i++){

c = b;

b = a;

a = a+c;

month++;

if(a >= n)

break;

}

printf("%d",month);

}

return 0;

}

**实验4-1-11 高空坠球 (20 分)**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int main()

{

double a,h,sum;

int n,i;

scanf("%lf %d",&h,&n);

sum = h;

if(n == 0){

printf("0.0 0.0");

exit(0);

}

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

a = h/pow(2,i - 1);

sum += a;

}

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

h = h/2;

printf("%.1lf %.1lf",sum,h);

return 0;

}

**实验4-1-12 黑洞数 (20 分)**

#include <stdio.h>

int max(int a,int b,int c);

int min(int a,int b,int c);

int main()

{

int i,j,n,t = 0;

int a[4];

scanf("%d",&n);

for(i = 1; ;i++){

t++;

a[1] = n%10;a[2] = n/10%10;a[3] = n/100;

n = max(a[1],a[2],a[3]) - min(a[1],a[2],a[3]);

if(n != 495)

printf("%d: %d - %d = %d\n",t,max(a[1],a[2],a[3]),min(a[1],a[2],a[3]),n);

else{

printf("%d: %d - %d = %d",t,max(a[1],a[2],a[3]),min(a[1],a[2],a[3]),n);

break;

}

}

return 0;

}

int max(int a,int b,int c)

{

int t,i,j;

int m[4]={0};

m[1]=a;m[2]=b;m[3]=c;

for(i = 1;i <= 3;i++){

for(j = 1;j <= 4-j;j++)

if(m[j]<m[j+1]){

t=m[j];m[j]=m[j+1];m[j+1]=t;

}

}

return m[1]\*100+m[2]\*10+m[3];

}

int min(int a,int b,int c)

{

int t,i,j;

int m[4]={0};

m[1]=a;m[2]=b;m[3]=c;

for(i = 1;i <= 3;i++){

for(j = 1;j <= 4-j;j++)

if(m[j]<m[j+1]){

t=m[j];m[j]=m[j+1];m[j+1]=t;

}

}

return m[3]\*100+m[2]\*10+m[1];

}

**实验4-2-2 求e的近似值 (15 分)**

#include <stdio.h>

double jie(int n);

int main()

{

int i,n;

double e = 1;

scanf("%d",&n);

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

e += 1/jie(i);

}

printf("%.8lf",e);

return 0;

}

double jie(int n)

{

int i;

double s = 1;

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

s \*= i;

return s;

}

**实验4-2-3 验证“哥德巴赫猜想” (20 分)**

#include <stdio.h>

#include <math.h>

int isPrime(int n);

int main()

{

int n,i,p,q;

scanf("%d",&n);

for(i = 2;i <= n/2;i++){

if(isPrime(i)&&isPrime(n-i)){

printf("%d = %d + %d",n,i,n-i);

break;

}

}

return 0;

}

int isPrime(int n)

{

int div;

for(div = 2;div <= sqrt(n);div++){

if(n%div == 0)

//break;

return 0;

}

return 1;

}

**实验4-2-4 换硬币 (20 分)**

#include <stdio.h>

int main()

{

int x,total = 0,count = 0; //x为零钱数额，total为硬币总数量，count为换法个数

int a,b,c;

scanf("%d",&x);

for(a = x/5;a > 0;a--)

for(b = x/2;b > 0;b--)

for(c = x;c > 0;c--)

if(a\*5+b\*2+c == x){

count++;

printf("fen5:%d, fen2:%d, fen1:%d, total:%d\n",a,b,c,a+b+c);

}

printf("count = %d",count);

return 0;

}

**实验4-2-5 水仙花数 (20 分)**

#include <stdio.h>

#include <math.h>

void fun(int ar[8],int m);

int main()

{

int n,j,k,s;

int sum,min,max,m;

int a[8]={0};

scanf("%d",&n);

min = pow(10,n - 1);

max = pow(10,n)-1;

for(j = min;j <= max;j++){

if(j > 5000000&&j< 9000000||j>2000000&&j<4000000) //我这一步只是为了解决n为7时的运行超时问题

continue;

m = j;

fun(a,m);

sum = 0;

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

sum += pow(a[k],n);

if(sum == j)

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

}

return 0;

}

void fun(int ar[8],int m)

{

int i = 1;

while(m){

ar[i] = m%10;

m = m/10;

if(m < 10)

ar[i+1] = m;

i++;

}

}

**实验4-2-6 输出三角形字符阵列 (15 分)**

#include <stdio.h>

int main()

{

int n,i;

char ch = 'A';

scanf("%d",&n);

do{

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

printf("%c ",ch);

ch++;

}

putchar('\n');

n--;

}while(n);

return 0;

}

**实验4-2-7 找完数 (20 分)**

#include <stdio.h>

void wanshu(int n);

int x = 0; //定义一个全局变量

int main()

{

int m,n,i,count = 0;

scanf("%d %d",&m,&n);

for(i = m;i <= n;i++){

wanshu(i);

}

if(x==0)

printf("None");

return 0;

}

void wanshu(int n)

{

int i,sum = 0,a[100] = {0},t = 0,s;

for(i = 1;i <= n/2;i++){

if(n%i == 0){

t++; //n最后会有t个因数

a[t] = i;

}

}

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

sum += a[i];

if(sum == n){

printf("%d = %d",n,a[1]);

for(s = 2;s <= t;s++)

printf(" + %d",a[s]);

putchar('\n');

x++;

}

}

**实验4-2-8 输出整数各位数字 (15 分)**

#include <stdio.h>

int main()

{

int n,j,i = 0,a[11] = {0};

scanf("%d",&n);

do

{

i++;

a[i] = n%10;

n = n/10;

}while(n);

for(j = i;j > 0;j--)

printf("%d ",a[j]);

return 0;

}

**实验4-2-9 梅森数 (20 分)**

#include <stdio.h>

#include <math.h>

int isPrime(int n); //素数判断函数

int main()

{

int n,i = 2,t;

scanf("%d",&n);

if(n < 2)

printf("None");

else{

do{

t = pow(2,i)-1;

if(isPrime(t)){ //如果i为素数

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

}

i++;

}while(i <= n);

}

return 0;

}

int isPrime(int n)

{

int div;

for(div = 2;div <= sqrt(n);div++){

if(n%div == 0)

return 0;

}

return 1;

}

**实验6-1 近似求PI (15 分)**

#include <stdio.h>

double fac(int n); //定义一个阶乘函数

int main()

{

int i = 0;

double eps,n = 0,s = 1,PI;

scanf("%le", &eps);

for(i = 1; ;i += 2)

{

s \*= i;

PI += fac(n)/s;

if(fac(n)/s < eps)

break;

n++;

}

printf("PI = %.5lf",2.0\*PI);

return 0;

}

double fac(int n)

{

int i;

double s = 1;

if(n == 0)

s = 1;

else

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

s \*= i;

return s;

}

**实验6-8 简单计算器 (20 分)**

#include <stdio.h>

int main()

{

int a,result;

int i,flag = 1;

char op;

scanf("%d",&a);

result = a;

for(i = 1; ;i++){

op = getchar();

scanf("%d",&a);

switch(op){

case '+':result += a;

break; //switch中的break只能跳出switch，不能跳出循环。

case '-':result -= a;

break;

case '\*':result \*= a;

break;

case '/':

if(a==0){

printf("ERROR");

flag = 0;

break;

}

else

result /= a;

break;

case '=':

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

flag = 0;

break;

default:

printf("ERROR");

flag = 0;

break;

}

if(flag == 0)

break;

}

return 0;

}

**实验6-9 统计一行文本的单词个数 (15 分)**

#include <stdio.h>

int main()

{

char ch[1000];

char c;

int i,word = 0,num = 0;

gets(ch);

for(i = 0;(c = ch[i]) != '\0';i++){

if(c == ' ')

word = 0;

else if(word == 0){

word = 1;

num++;

}

}

printf("%d",num);

return 0;

}

**实验7-1-1 简化的插入排序 (15 分)**

#include <stdio.h>

int main()

{

int n,i,j,b,temp;

int a[11];

scanf("%d",&n);

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

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

scanf("%d",&b);

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

if(b <= a[i]){

for(j = n;j > i;j--){

a[j] = a[j-1];

}

a[i] = b;

break;

}

if(b == a[n-1])

a[n] = b;

if(b>a[n-1])

a[n] = b;

}

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

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

return 0;

}

**实验7-1-2 求最大值及其下标 (20 分)**

#include <stdio.h>

int main()

{

int i,n,max,j = 0,a[10];

scanf("%d",&n);

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

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

max = a[0];

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

if(max<a[i]){

max = a[i];

j = i;

}

}

printf("%d %d",max,j);

return 0;

}

**实验7-1-3 将数组中的数逆序存放 (20 分)**

#include <stdio.h>

int main()

{

int i,n,a[10],b[10];

scanf("%d",&n);

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

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

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

b[n-i-1]=a[i];

}

printf("%d",b[0]);

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

printf(" %d",b[i]);

return 0;

}

**实验7-1-4 找出不是两个数组共有的元素 (20 分)**

#include<stdio.h>

int main()

{

int a[20], b[20], c[20];

int m, n, i, j, k=0;

scanf("%d", &m);

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

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

scanf("%d", &n);

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

scanf("%d", &b[i]);

for(i=0; i<m; i++){

for(j=0; j<n; j++){

if(a[i]==b[j])

break;

}

if(j>=n){

c[k]=a[i];

k++;

}

}

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

for(j=0; j<m; j++){

if(b[i]==a[j])

break;

}

if(j>=m){

c[k]=b[i];

k++;

}

}

printf("%d", c[0]);

for(i=1; i<k; i++){

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

if(c[i]==c[j])

break;

}

if(j>=i)

printf(" %d", c[i]);

}

return 0;

}

**实验7-1-5 选择法排序 (20 分)**

#include <stdio.h>

int main()

{

int n,i,j,temp;

int a[10];

scanf("%d",&n);

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

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

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

{

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

{

if(a[i]<a[j])

{

temp=a[i];

a[i]=a[j];

a[j]=temp;

}

}

}

printf("%d",a[0]);

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

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

//printf("%c",8);

//printf("xxx");

return 0;

}

**实验7-1-6 求一批整数中出现最多的个位数字 (20 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

int fmax(int a[]);

int main()

{

int n, i, a[1000], fqy[10] = { 0 };

int dight,max\_fqy;

scanf("%d",&n);

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

{

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

}

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

{

while (a[i] > 0)

{

dight = a[i] % 10;

fqy[dight]++;

a[i] /= 10;

}

}

max\_fqy = fmax(fqy);

printf("%d:",max\_fqy);

for (i = 0; i < 10; i++)

{

if (fqy[i] == max\_fqy)

printf(" %d",i);

}

return 0;

}

int fmax(int a[])

{

int i,max;

max = a[0];

for (i = 0; i < 10; i++)

if (a[i] > max)

max = a[i];

return max;

}

**实验7-1-7 查找整数 (10 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

int fmax(int a[]);

int main()

{

int n, x, i, a[20], count = 0;

scanf("%d %d",&n,&x);

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

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

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

{

if (a[i] == x)

{

printf("%d",i);

count++;

}

}

if (count == 0)

printf("Not Found");

return 0;

}

**实验7-1-8 输出数组元素 (15 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

int main()

{

int i, n, a[10] = { 0 }, count = 1;

scanf("%d",&n);

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

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

printf("%d",a[1]-a[0]);

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

{

printf(" %d", a[i + 1] - a[i]);

count++;

if (count == 3)

{

putchar('\n');

if(i + 1 != n-1)

{

printf("%d", a[i + 2] - a[i + 1]);

count=1;

}

i++;

}

}

return 0;

}

**实验7-1-9 数字加密 (15 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

int main()

{

int i, x, a[4];

scanf("%d", &x);

a[0] = (x / 1000 + 9) % 10; //千位

a[1] = (x / 100 % 10 + 9) % 10; //百位

a[2] = (x / 10 % 10 + 9) % 10; //十位

a[3] = (x % 10 + 9) % 10; //个位

printf("The encrypted number is %d%d%d%d", a[2], a[3], a[0], a[1]);

return 0;

}

**实验7-1-10 交换最小值和最大值 (15 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

#include <stdlib.h>

int main()

{

int i, n, a[10], max, min, x, y, temp;

scanf("%d", &n);

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

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

if(n == 1)

{

printf("%d ", a[0]);

exit(0);

}

min = a[0];

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

if (min > a[i])

{

min = a[i];

y = i;

}

a[y] = a[0];

a[0] = min;

max = a[0];

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

if(max < a[i])

{

max = a[i];

x = i;

}

a[x] = a[n - 1];

a[n - 1] = max;

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

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

return 0;

}

**实验7-1-11 求整数序列中出现次数最多的数 (15 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

#include <math.h>

int main()

{

int n, a[1000], b[1000], count = 0;

int temp, max;

scanf("%d", &n);

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

{

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

b[i] = a[i];

}

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

{

for (int j = 0; j < n; j++)

{

if (a[i] == b[j])

count++;

}

if (i == 0)

{

temp = count;

max = a[i];

}

else if(temp < count)

{

temp = count;

max = a[i];

}

count = 0;

}

printf("%d %d",max,temp);

return 0;

}

**实验7-1-12 组个最小数 (20 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

int main()

{

int i, a[10];

for (i = 0; i < 10; i++)

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

for (i = 1; i < 10; i++)

{

if (a[i] != 0)

{

printf("%d", i);

a[i]--;

break;

}

}

for (i = 0; i < 10; i++)

{

while (a[i] != 0)

{

printf("%d", i);

a[i]--;

}

}

return 0;

}

**实验7-1-13 装箱问题 (20 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

#include <string.h>

int main()

{

int n, s[1001], a[1001], max = 1; //定义物品个数n个

scanf("%d", &n);

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

scanf("%d", &s[i]); //将每个物品大小存入数组中

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

a[i] = 100; //将数组a赋值为100

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

{

for (int j = 1; j <= n; j++)

{

if (s[i] <= a[j])

{

if (j > max)

max = j; //记录最大箱子号

printf("%d %d\n", s[i], j);

a[j] = a[j] - s[i]; //更新数组a

break;

}

}

}

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

return 0;

}

**实验7-2-1 求矩阵各行元素之和 (15 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

int main()

{

int m, n, a[6], s = 0;

scanf("%d %d", &m, &n);

for (int i = 0; i < m; i++)

{

for (int j = 0; j < n; j++)

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

for (int j = 0; j < n; j++)

s += a[j];

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

s = 0;

}

return 0;

}

**实验7-2-2 矩阵运算 (20 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

int main()

{

int n, sum = 0, a[10][10];

scanf("%d", &n);

for (int i = 0; i < n; i++) //按格式输入矩阵

for (int j = 0; j < n; j++)

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

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

for (int j = 0; j < n; j++)

{

if (!(i + j == n - 1 || j == n - 1 || i == n - 1))

//a[i][j] = 0; //如果符合题意，就将其赋值为0

sum += a[i][j]; //求和

}

printf("%d", sum);

return 0;

}

**实验7-2-3 求矩阵的局部极大值 (15 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

int main()

{

int m, n, count = 0, a[100][21];

scanf("%d %d", &m, &n);

for (int i = 1; i <= m; i++) //按照格式输入矩阵

for (int j = 1; j <= n; j++)

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

for (int i = 2; i < m; i++) //从第2行开始，到倒数第一行结束

{

for (int j = 2; j < n; j++) //从第2列开始，到倒数第一列结束

{

if (a[i][j] > a[i + 1][j] && a[i][j] > a[i - 1][j] &&

a[i][j] > a[i][j + 1] && a[i][j] > a[i][j - 1])//判断是否为矩阵的局部极大值

{

printf("%d %d %d\n", a[i][j], i, j);

count++;

}

}

}

if (count == 0)

printf("None %d %d", m, n);

return 0;

}

**实验7-2-4 计算天数 (15 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

int main()

{

int year, month, days;

scanf("%d/%d/%d", &year, &month, &days);

for (int i = 1; i <= month - 1; i++) //利用循环，每次遍历一个月份，就加上该月份的天数

{

switch (i)

{

case 1:

case 3:

case 5:

case 7:

case 8:

case 10:

case 12:days += 31;

break;

case 2:days += 28;

break;

case 4:

case 6:

case 9:

case 11:days += 30;

break;

}

}

if (year % 4 == 0 && year % 100 != 0 || year % 400 == 0) //判断是否为闰年

{

if (month > 2) //月份大于2，就加上闰年多出的一天

printf("%d", days+1);

else

printf("%d", days);

}

else

printf("%d", days);

return 0;

}

**实验7-2-5 判断上三角矩阵 (15 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

int main()

{

int t, n, count = 0, a[10][10];

scanf("%d", &t);

for (int k = 0; k < t; k++)

{

scanf("%d", &n);

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

for (int j = 0; j < n; j++)

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

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

for (int j = i - 1; j >= 0; j--)

if (a[i][j] != 0)

count = 1;

if (count == 0)

printf("YES\n");

else

printf("NO\n");

count = 0;

}

return 0;

}

**实验7-2-6 打印杨辉三角 (20 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

int main()

{

int t, n, a[10][10];

scanf("%d", &n);

t = n - 1;

for (int i = 0; i < n; i++) //先给每行第一列数和最后一个数赋值1

a[i][0] = a[i][i] = 1;

for (int i = 2; i < n; i++) //利用循环给其他数赋值

for (int j = 1; j < i; j++)

a[i][j] = a[i - 1][j] + a[i - 1][j - 1];

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

{

for (int k = 0; k < t; k++) //先输出空格

printf(" ");

t--;

for (int j = 0; j < i + 1; j++) //输出数字+空格

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

printf("\n");

}

return 0;

}

**实验7-2-7 方阵循环右移 (20 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

int main()

{

int m, n, a[6][6];

scanf("%d %d", &m, &n);

m = m % n; //考虑到m可能大于n，这里取模

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

for (int j = 0; j < n; j++)

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

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

{

for (int j = n-m; j < n; j++) //先输出右移后超出最后一列的数

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

for (int j = 0; j < n - m; j++) //在输出右移后没有超过最后一列的数

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

printf("\n");

}

return 0;

}

**实验7-2-8 找鞍点 (20 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

int is\_line\_max(int n, int x, int i, int a[6][6]);//判断是否为行最大值

int is\_rank\_min(int n, int x, int j, int a[6][6]);//判断是否为列最小值

int main()

{

int n, flag = 0, a[6][6];

scanf("%d", &n);

for (int i = 0; i < n; i++) //按格式输入

for (int j = 0; j < n; j++)

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

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

for (int j = 0; j < n; j++)

if (is\_line\_max(n, a[i][j], i, a))

if (is\_rank\_min(n, a[i][j], j, a))

{

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

flag = 1; //如果是鞍点，就做标记

}

if (flag == 0)

printf("NONE");

return 0;

}

int is\_line\_max(int n, int x, int i, int a[6][6])

{

for (int k = 0; k < n; k++)

if (x < a[i][k])

return 0;

return 1;

}

int is\_rank\_min(int n, int x, int j, int a[6][6])

{

for (int k = 0; k < n; k++)

if (x > a[k][j])

return 0;

return 1;

}

**实验7-2-9 螺旋方阵 (20 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

//参考刘汝佳"蛇形填数"的算法

#include <stdio.h>

#include <string.h> //内含memset()函数

int main()

{

int n, x, y, tot = 0, a[10][10];

scanf("%d", &n);

memset(a, 0, sizeof(a)); //将数组元素全部初始化为0,以便之后判断

tot = a[x = 0][y = 0] = 1;

while (tot < n \* n)

{ //先进性预判，碰壁就不往下走，!a[x][y+1]等价于a[x][y+1]==0

while (y + 1 < n && !a[x][y + 1]) a[x][++y] = ++tot;

while (x + 1 < n && !a[x + 1][y]) a[++x][y] = ++tot;

while (y - 1 >= 0 && !a[x][y - 1]) a[x][--y] = ++tot;

while (x - 1 >= 0 && !a[x - 1][y]) a[--x][y] = ++tot;

}

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

{

for (int j = 0; j < n; j++)

printf("%3d", a[i][j]);

printf("\n");

}

return 0;

}

**实验7-2-10 简易连连看 (20 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

#include <stdlib.h> //内含exit()函数

void print(int n, char a[10][10]); //输出字符串数组函数

int main()

{

int n, k, count = 0, fail = 0;

char a[10][10];

scanf("%d", &n);

for (int i = 1; i <= 2 \* n; i++)

for (int j = 1; j <= 2 \* n; j++)

{

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

if (a[i][j] == ' ' || a[i][j] == '\n') //过滤无效字符

j--;

}

scanf("%d", &k);

for (int i = 1; i <= k; i++)

{

int x1, x2, y1, y2;

scanf("%d %d %d %d", &x1, &y1, &x2, &y2);

if (a[x1][y1] == a[x2][y2] && a[x1][y1] != '\*')

{

a[x1][y1] = a[x2][y2] = '\*';

count += 2;

if (count != 2 \* n \* 2 \* n)

print(n, a);

}

else if (a[x1][y1] != a[x2][y2] || a[x1][y1] == '\*' && a[x2][y2] == '\*')

{

fail++;

printf("Uh-oh\n");

}

if (count == 2 \* n \* 2 \* n)

{

printf("Congratulations!");

exit(0); //直接退出程序

}

if (fail == 3)

{

printf("Game Over");

exit(0);

}

}

return 0;

}

void print(int n, char a[10][10])

{

for (int i = 1; i <= 2 \* n; i++)

{

for (int j = 1; j <= 2 \* n; j++)

{

if (j == 1)

printf("%c", a[i][j]);

else

printf(" %c", a[i][j]);

}

printf("\n");

}

}

**实验7-3-1 字符串逆序 (15 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

#include <string.h>

int main()

{

char a[81];

gets(a);

for (int i = strlen(a) - 1; i >= 0; i--)

printf("%c", a[i]);

return 0;

}

**实验7-3-2 查找指定字符 (15 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

#include <string.h>

int main()

{

char ch, a[81];

int flag = 0;

ch = getchar();

getchar();

gets(a);

for (int i = strlen(a) - 1; i >= 0; i--)

if (a[i] == ch)

{

printf("index = %d", i);

flag = 1;

break;

}

if (flag == 0)

printf("Not Found");

return 0;

}

**实验7-3-3 统计大写辅音字母 (15 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

#include <string.h>

int main()

{

char c[81];

int count = 0;

gets(c);

for (int i = strlen(c) - 1; i >= 0; i--)

if (c[i] >= 'A' && c[i] <= 'Z')

if (c[i] != 'A' && c[i] != 'E' && c[i] != 'I' && c[i] != 'O' && c[i] != 'U')

count++;

printf("%d", count);

return 0;

}

**实验7-3-4 字符串替换 (15 分)**

#include <stdio.h>

int main()

{

char ch[81];

char c;

int i;

gets(ch);

for(i = 0;(c=ch[i])!='\0';i++)

{

if(c>='A'&&c<='Z')

{

c='Z'-(c-'A');

printf("%c",c);

}

else

printf("%c",c);

}

return 0;

}

**实验7-3-5 输出大写英文字母 (15 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

#include <string.h>

int main()

{

char a[81], c[81];

int flag1 = 0, flag2 = 0, count = 0;

gets(a);

for (int i = 0; i < strlen(a); i++)

if (a[i] >= 'A' && a[i] <= 'Z')

{

flag2 = 1;

for (int j = 0; j < count; j++)

if (c[j] == a[i])

flag1 = 1;

if (flag1 == 0)

{

c[count] = a[i];

printf("%c", a[i]);

count++;

}

flag1 = 0;

}

if (flag2 == 0)

printf("Not Found");

return 0;

}

**实验7-3-6 字符串转换成十进制整数 (15 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

#include <string.h>

#include <ctype.h> //包含toupper()函数

int main()

{

char c[100], ch[100] = { '\0' };

int j, sum, n;

j = sum = 0;

gets(c);

for (int i = 0; i < strlen(c); i++)

{

if (c[i] == '#')

break;

else if (c[i]=='-'||c[i] >= '0' && c[i] <= '9' || toupper(c[i]) >= 'A' && toupper(c[i]) <= 'F')

{

ch[j] = c[i];

j++;

}

}

for (int i = 0; i < strlen(ch); i++)

{

if (ch[i] != '-')

{

if (ch[i] >= 'A' && ch[i] <= 'F')

n = ch[i] - 'A' + 10;

else if (ch[i] >= 'a' && ch[i] <= 'f')

n = ch[i] - 'a' + 10;

else

n = ch[i] - '0';

sum = sum \* 16 + n;

}

}

if (ch[0] == '-' && sum != 0)

printf("-%d", sum);

else

printf("%d", sum);

return 0;

}

**实验7-3-7 字符转换 (15 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

#include <string.h>

int main()

{

char c[81];

int n = 1, sum = 0;

gets(c);

for (int i = strlen(c); i >= 0; i--)

if (c[i] >= '0' && c[i] <= '9')

{

sum += n \* (c[i] - 48);

n \*= 10;

}

printf("%d", sum);

return 0;

}

**实验7-3-8 统计字符出现次数 (20 分)**

#include <stdio.h>

int main()

{

char ch[81];

char c,s;

int i,count=0;

gets(ch);

c=getchar();

for(i = 0; (s=ch[i])!='\0'; i++)

{

if(c==s)

count++;

}

printf("%d",count);

return 0;

}

**实验7-3-9 字符串字母大小写转换 (15 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

#include <string.h>

int main()

{

char ch;

while ((ch = getchar()) != '#')

{

if (ch >= 'a' && ch <= 'z')

printf("%c", ch - 32);

else if (ch >= 'A' && ch <= 'Z')

printf("%c", ch + 32);

else

printf("%c", ch);

}

return 0;

}

**实验7-3-10 删除重复字符 (20 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

#include <string.h>

#define maxn 81

int main()

{

char c[maxn], ch[maxn] = { '\0' }, temp;

int flag = 1, index = 0;

gets(c);

for (int i = 0; i < strlen(c); i++)

{

for (int j = 0; j < strlen(ch); j++)

if (c[i] == ch[j])

flag = 0;

if (flag)

{

ch[index] = c[i];

index++;

}

flag = 1;

}

//排序

for (int i = 0; i < strlen(ch); i++)

for (int j = i; j < strlen(ch); j++)

if (ch[i] > ch[j])

{

temp = ch[i];

ch[i] = ch[j];

ch[j] = temp;

}

puts(ch);

return 0;

}

**实验8-1-9 输出学生成绩 (20 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

#include <stdlib.h> //为malloc()、free()提供原型

int main()

{

int n;

double\* p, max = -1, min = 101, ave, sum = 0;

scanf("%d", &n);

p = (double\*)malloc(n \* sizeof(double));

if (p == NULL)

exit(EXIT\_FAILURE);

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

{

scanf("%lf", &p[i]);

sum += p[i];

if (p[i] > max)

max = p[i];

if (p[i] < min)

min = p[i];

}

printf("average = %.2f\n", sum / n);

printf("max = %.2f\n", max);

printf("min = %.2f\n", min);

free(p);

return 0;

}

**实验8-2-1 找最小的字符串 (15 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

#include <string.h>

int main()

{

int n;

char ch[81], min[81];

scanf("%d", &n);

scanf("%s", ch);

strcpy(min, ch);

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

{

scanf("%s", ch);

if (strcmp(min, ch) > 0)

strcpy(min, ch);

}

printf("Min is: %s", min);

return 0;

}

**实验8-2-2 找最长的字符串 (15 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

#include <string.h>

int main()

{

int n;

char ch[81], max[81];

scanf("%d", &n);

scanf("%s", ch);

strcpy(max, ch);

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

{

scanf("%s", ch);

if (strlen(max) < strlen(ch))

strcpy(max, ch);

}

printf("The longest is: %s", max);

return 0;

}

**实验8-2-8 字符串排序 (20 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

#include <string.h>

int main()

{

char c[5][81];

char ch[81];

scanf("%s %s %s %s %s", &c[0], &c[1], &c[2], &c[3], &c[4]);

for(int i=0;i<5;i++)

for(int j=i;j<5;j++)

if (strcmp(c[i],c[j])>0)

{

strcpy(ch, c[j]);

strcpy(c[j], c[i]);

strcpy(c[i], ch);

}

printf("After sorted:\n");

for (int i = 0; i < 5; i++)

puts(c[i]);

return 0;

}

**实验8-2-10 IP地址转换 (20 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

#include <math.h>

int main()

{

char c;

for (int i = 0; i < 4; i++)

{

int n = 7, sum = 0;

for (int j = 0; j < 8; j++)

{

c = getchar();

sum += (c - 48) \* pow(2, n);

n--;

}

if (i < 3)

printf("%d.", sum);

else

printf("%d", sum);

}

return 0;

}

**实验9-1 计算职工工资 (15 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

int main()

{

int n;

scanf("%d", &n);

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

{

char staff[10];

float a, b, c;

scanf("%s %f %f %f", &staff, &a, &b, &c);

printf("%s %.2f\n", staff, a + b - c);

}

return 0;

}

**实验9-2 时间换算 (15 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

int main()

{

int h, m, s, n;

scanf("%d:%d:%d", &h, &m, &s);

scanf("%d", &n);

if (s + n >= 60)

{

m++;

s += n - 60;

}

else

s += n;

if (m >= 60)

{

h++;

m -= 60;

}

if (h >= 24)

h -= 24;

printf("%02d:%02d:%02d", h, m, s);

return 0;

}

**实验9-3 计算平均成绩 (15 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

struct student

{

char num[6]; //学号

char name[11]; //姓名

float score; //分数

}s[10];

int main()

{

int n;

float sum = 0;

scanf("%d", &n);

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

{

scanf("%s %s %f", s[i].num, s[i].name, &s[i].score);

sum += s[i].score;

}

printf("%.2f\n", sum / n);

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

if (s[i].score < sum / n)

printf("%s %s\n", s[i].name, s[i].num);

return 0;

}

**实验9-5 查找书籍 (20 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

struct book

{

char name[32]; //书名

double price; //价格

}s[10], high, low;

int main()

{

int n;

scanf("%d", &n);

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

{

scanf("\n"); //每次输入数字后都会有一个'\n'，用scanf吞掉它

gets(s[i].name);

scanf("%lf",&s[i].price);

}

high = low = s[0];

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

{

if (s[i].price > high.price)

high = s[i];

if (s[i].price < low.price)

low = s[i];

}

printf("%.2lf, %s\n", high.price, high.name);

printf("%.2lf, %s\n", low.price, low.name);

return 0;

}

**实验9-7 找出总分最高的学生 (15 分)**

#include <stdio.h>

struct student {

char num[6]; //学号

char name[11]; //姓名

int a,b,c; //三门成绩

}s[10];

int main()

{

int n,i,j = 0,max; //一定要给j初始化，不然当第一个学生为最大成绩时，会出现错误

scanf("%d",&n);

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

scanf("%s %s %d %d %d",s[i].num,s[i].name,&s[i].a,&s[i].b,&s[i].c);

max = s[0].a + s[0].b + s[0].c;

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

if(max < s[i].a + s[i].b + s[i].c){

max = s[i].a + s[i].b + s[i].c;

j = i;

}

}

printf("%s %s %d",s[j].name,s[j].num,max);

return 0;

}

**实验9-8 通讯录排序 (20 分)**

#include <stdio.h>

struct fri{ //结构声明

char name[11]; //姓名字符串数组

char num[18]; //电话号码字符串数组

int both; //生日

}s[10],temp;

int main()

{

int n,i,j;

scanf("%d",&n);

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

scanf("%s %d %s",s[i].name,&s[i].both,s[i].num);

}

for(i = 0;i < n - 1;i++) //选择法排序

for(j = i;j < n;j++){

if(s[i].both > s[j].both){

temp = s[i];

s[i] = s[j];

s[j] = temp;

}

}

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

printf("%s %d %s\n",s[i].name,s[i].both,s[i].num);

return 0;

}

**实验9-9 有理数比较 (10 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

int main()

{

int a1, b1, a2, b2;

scanf("%d/%d %d/%d", &a1, &b1, &a2, &b2);

if ((float)a1 / b1 > (float)a2 / b2)

printf("%d/%d > %d/%d", a1, b1, a2, b2);

if ((float)a1 / b1 < (float)a2 / b2)

printf("%d/%d < %d/%d", a1, b1, a2, b2);

if ((float)a1 / b1 == (float)a2 / b2)

printf("%d/%d = %d/%d", a1, b1, a2, b2);

return 0;

}

**实验9-10 平面向量加法 (15 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

int main()

{

double x1, y1, x2, y2, x, y;

scanf("%lf %lf %lf %lf", &x1, &y1, &x2, &y2);

x = x1 + x2; y = y1 + y2;

if (x > -0.05 && x < 0)

x = 0;

if (y > -0.05 && y < 0)

y = 0;

printf("(%.1lf, %.1lf)", x, y);

return 0;

}

**实验10-1 圆形体体积计算器 (20 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

#include <stdlib.h> //包含exit()函数

#define PI 3.141592653579 //π

void menu(); //菜单栏

double ball(double r); //V-球

double cylinder(double r, double h); //V-圆柱

double cone(double r, double h); //V-圆锥

int main()

{

int n;

double r, h;

while (1)

{

menu();

scanf("%d", &n);

switch (n)

{

case 1:

printf("Please enter the radius:\n");

scanf("%lf", &r);

printf("%.2lf\n", ball(r));

break;

case 2:

printf("Please enter the radius and the height:\n");

scanf("%lf %lf", &r, &h);

printf("%.2lf\n", cylinder(r, h));

break;

case 3:

printf("Please enter the radius and the height:\n");

scanf("%lf %lf", &r, &h);

printf("%.2lf\n", cone(r, h));

break;

default:

exit(0); //直接退出

}

}

return 0;

}

void menu()

{

printf("1-Ball\n");

printf("2-Cylinder\n");

printf("3-Cone\n");

printf("other-Exit\n");

printf("Please enter your command:\n");

}

double ball(double r)

{

return ((double)4 / 3) \* PI \* r \* r \* r;

}

double cylinder(double r, double h)

{

return PI \* r \* r \* h;

}

double cone(double r, double h)

{

return ((double)1 / 3) \* PI \* r \* r \* h;

}

**实验11-1-1 英文单词排序 (25 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

#include <string.h>

int main()

{

char word[20][10];

char temp[10];

int index = 0;

gets(word[index]);

while ((word[index][0]) != '#')

{

index++;

gets(word[index]);

}

for (int i = 0; i < index; i++)

for (int j = i; j < index; j++)

{

if (strlen(word[i]) > strlen(word[j]))

{

strcpy(temp, word[i]);

strcpy(word[i], word[j]);

strcpy(word[j], temp);

}

}

for (int i = 0; i < index; i++)

printf("%s ", word[i]);

return 0;

}

**实验11-1-7 藏头诗 (15 分)**

#include <stdio.h>

int main()

{

char a[4][15],b[9];

int i,j=0;

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

scanf("%s",a[i]);

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

{

b[j++]=a[i][0];

b[j++]=a[i][1];

}

b[j]='\0';

printf("%s",b);

return 0;

}

**实验11-1-9 藏尾诗 (20 分)**

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

#include <string.h>

int main()

{

char a[4][20], b[9];

int j = 0;

for (int i = 0; i < 4; i++)

scanf("%s", a[i]);

for (int i = 0; i < 4; i++)

{

b[j++] = a[i][strlen(a[i]) - 2];

b[j++] = a[i][strlen(a[i]) - 1];

}

b[j] = '\0';

printf("%s", b);

return 0;

}

**函数题**

**实验2-4-1 统计各位数字之和是5的数 (20 分)**

int is(int number)

{

int sum = number % 10;

do

{

number /= 10;

sum += number % 10;

}while (number / 10 > 0);

if (sum == 5)

return 1;

else

return 0;

}

void count\_sum(int a, int b)

{

int count, sum;

count = sum = 0;

for (int i = a; i <= b; i++)

if (is(i))

{

count++;

sum += i;

}

printf("count = %d, sum = %d", count, sum);

}

**实验2-4-5 简单实现x的n次方 (10 分)**

double mypow( double x, int n )

{

double y = 1;

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

y \*= x;

return y;

}

**实验5-1 使用函数计算两个复数之积 (10 分)**

void complex\_prod(double x1, double y1, double x2, double y2)

{

result\_real = x1 \* x2 - y1 \* y2;

result\_imag = x1 \* y2 + x2 \* y1;

}

**实验5-2 符号函数 (10 分)**

int sign( int x )

{

int y;

if(x > 0)

y = 1;

else if(x == 0)

y = 0;

else

y = -1;

return y;

}

**实验5-3 使用函数求奇数和 (15 分)**

int even(int n)

{

if (n % 2 == 0)

return 1;

return 0;

}

int OddSum(int List[], int N)

{

int sum = 0;

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

if (even(List[i]) == 0)

sum += List[i];

return sum;

}

**实验5-4 使用函数计算两点间的距离 (10 分)**

double dist( double x1, double y1, double x2, double y2 )

{

return sqrt((x1-x2)\*(x1-x2)+(y1-y2)\*(y1-y2));

}

**实验5-5 使用函数求素数和 (20 分)**

int prime(int p)

{

if (p <= 1)

return 0;

for (int div = 2; div <= sqrt(p); div++)

if (p % div == 0)

return 0;

return 1;

}

int PrimeSum(int m, int n)

{

int sum = 0;

for (int i = m; i <= n; i++)

if (prime(i))

sum += i;

return sum;

}

**实验5-6 使用函数判断完全平方数 (10 分)**

int IsSquare( int n )

{

int m = floor(sqrt(n) + 0.5);

if(m \* m == n)

return 1;

return 0;

}

**实验5-7 使用函数求1到10的阶乘和 (10 分)**

double fact( int n )

{

int s = 1;

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

s \*= i;

return s;

}

**实验5-8 使用函数统计指定数字的个数 (15 分)**

int CountDigit( int number, int digit )

{

int count = 0;

if(number == 0 && digit == 0)

count = 1;

if(number < 0)

number = -number;

while(number > 0)

{

if(number % 10 == digit)

count++;

number /= 10;

}

return count;

}

**实验5-9 使用函数输出水仙花数 (20 分)**

int pow(int x, int n)

{

int s = 1;

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

s \*= x;

return s;

}

int narcissistic(int number)

{

int n = 0, count = 0, a[5];

int num = number;

for (int i = 0;; i++)

{

a[i] = number % 10;

count++;

if (number / 10 <= 0)

break;

number /= 10;

}

for (int i = 0; i < count; i++)

n += pow(a[i], count);

if (n == num)

return 1;

return 0;

}

void PrintN(int m, int n)

{

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

if (narcissistic(i))

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

}

**实验5-10 使用函数求余弦函数的近似值 (15 分)**

double fac(int n)

{

double s = 1;

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

s \*= i;

return s;

}

double funcos(double e, double x)

{

double sum = 1;

for (int i = 2, j = 1; ; i += 2, j++)

{

if (j % 2 != 0)

sum -= pow(x, i) / fac(i);

else

sum += pow(x, i) / fac(i);

if (pow(x, i) / fac(i) < e)

break;

}

return sum;

}

**实验5-11 使用函数求最大公约数 (10 分)**

int gcd( int x, int y )

{

int temp;

if(x > y)

{

temp = x;

x = y;

y = temp;

}

do

{

temp = x % y;

x = y;

y = temp;

}while(temp != 0);

return x;

}

**实验6-2 分类统计字符个数 (15 分)**

void StringCount(char s[])

{

int letter = 0, blank = 0, digit = 0, other = 0;

int i = 0;

do

{

if (s[i] >= 'a' && s[i] <= 'z'|| s[i] >= 'A' && s[i] <= 'Z')

letter++;

else if (s[i] == ' ' || s[i] == '\n')

blank++;

else if (s[i] >= '0' && s[i] <= '9')

digit++;

else

other++;

i++;

}while(s[i] != '\0');

printf("letter = %d, blank = %d, digit = %d, other = %d", letter, blank, digit, other);

}

**实验6-3 使用函数求特殊a串数列和 (20 分)**

int fn( int a, int n )

{

int r = a;

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

a = (a\*10)+r;

return a;

}

int SumA( int a, int n )

{

int sum = 0;

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

sum += fn(a, i);

return sum;

}

**实验6-4 使用函数输出指定范围内的完数 (20 分)**

int factorsum(int number)

{

int sum = 0, j = 0, a[100];

for (int i = 1; i <= number / 2; i++)

if (number % i == 0)

a[j++] = i;

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

sum += a[i];

if (sum == number)

return number;

return 0;

}

void PrintPN(int m, int n)

{

int flag = 0, a[100];

for (int i = m; i <= n; i++)

{

int k = 0;

if (factorsum(i))

{

flag = 1;

for (int j = 1; j <= i / 2; j++)

if (i % j == 0)

a[k++] = j;

printf("%d = %d", i, a[0]);

}

for (int x = 1; x < k; x++)

printf(" + %d", a[x]);

if (factorsum(i))

putchar('\n');

}

if (!flag)

printf("No perfect number");

}

**实验6-5 使用函数输出指定范围内的Fibonacci数 (20 分)**

int fib(int n)

{

if (n <= 2)

return 1;

return fib(n - 1) + fib(n - 2);

}

void PrintFN(int m, int n)

{

int a[100], j = 0, flag = 0;

for (int i = 1; i <= 20; i++)

if (m <= fib(i) && n >= fib(i))

{

a[j++] = fib(i);

flag = 1;

}

if (flag)

{

printf("%d", a[0]);

for (int i = 1; i < j; i++)

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

}

else

printf("No Fibonacci number");

}

**实验6-6 使用函数验证哥德巴赫猜想 (20 分)**

int prime(int p)

{

if (p == 1)

return 0;

for (int div = 2; div <= sqrt(p); div++)

if (p % div == 0)

return 0;

return 1;

}

void Goldbach(int n)

{

for (int i = 2; i <= n; i++)

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

{

printf("%d=%d+%d", n, i, n - i);

break;

}

}

**实验6-7 使用函数输出一个整数的逆序数 (20 分)**

int reverse( int number )

{

if (number < 0)

{

printf("-");

number = -number;

}

int n = 0;

do

{

n = n \* 10 + number % 10 \* 10;

number /= 10;

}while(number);

return n / 10;

}

**实验8-1-1 利用指针找最大值 (10 分)**

void findmax( int \*px, int \*py, int \*pmax )

{

if (\*px > \*py)

\*pmax = \*px;

else

\*pmax = \*py;

}

**实验8-1-2 计算两数的和与差 (10 分)**

void sum\_diff( float op1, float op2, float \*psum, float \*pdiff )

{

\*psum = op1 + op2;

\*pdiff = op1 - op2;

}

**实验8-1-3 拆分实数的整数与小数部分 (15 分)**

void splitfloat( float x, int \*intpart, float \*fracpart )

{

\*intpart = (int)x;

\*fracpart = x - \*intpart;

}

**实验8-1-4 使用函数的选择法排序 (25 分)**

void sort( int a[], int n )

{

int temp;

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

for (int j = i; j < n; j++)

if (a[i] > a[j])

{

temp = a[i];

a[i] = a[j];

a[j] = temp;

}

}

**实验8-1-5 在数组中查找指定元素 (15 分)**

int search( int list[], int n, int x )

{

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

if (list[i] == x)

return i;

return -1;

}

void f( char \*p )

{

int index = 0;

while (p[index] != '\0')

index++;

for (int i = 0; i < index / 2; i++)

{

char temp;

temp = p[i];

p[i] = p[index - 1 - i];

p[index - 1 - i] = temp;

}

}

**实验8-1-7 数组循环右移 (20 分)**

int ArrayShift(int a[], int n, int m)

{

int b[MAXN], j = 0;

m = m % n;

for (int i = n - m; i < n; i++)

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

for (int i = 0; i < n - m; i++)

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

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

a[i] = b[i];

return a;

}

**实验8-1-8 报数 (20 分)**

void CountOff(int n, int m, int out[])

{

int a[MAXN], count = 1, index = 1;

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

a[i] = i + 1;

while (count <= n)

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

{

if (a[i] != 0)

if (index % m == 0)

{

out[i] = count++;

a[i] = 0;

index = 1;

}

else

index++;

}

}

**实验8-2-3 删除字符 (20 分)**

void delchar(char\* str, char c)

{

int i = 0, j = 0;

while (str[i] != '\0')

{

if (str[i] != c)

str[j++] = str[i];

i++;

}

str[j] = '\0';

return str;

}

**实验8-2-4 使用函数实现字符串部分复制 (20 分)**

void strmcpy(char\* t, int m, char\* s)

{

int i = m - 1, j = 0;

while (t[i] != '\0')

{

s[j++] = t[i];

i++;

}

s[j] = '\0';

return s;

}

**实验8-2-5 判断回文字符串 (20 分)**

bool palindrome(char\* s)

{

int j = 0;

char str[MAXN+1];

for (int i = strlen(s) - 1; i >= 0; i--)

str[j++] = s[i];

str[j]='\0';

for (int i = 0; i < strlen(s); i++)

if (s[i] != str[i])

return false;

return true;

}

**实验8-2-6 分类统计各类字符个数 (15 分)**

void StringCount( char \*s )

{

int capital = 0, minuscule = 0, blank = 0, digit = 0, other = 0;

int i = 0;

do

{

if (s[i] >= 'A' && s[i] <= 'Z')

capital++;

else if (s[i] >= 'a' && s[i] <= 'z')

minuscule++;

else if (s[i] == ' ')

blank++;

else if (s[i] >= '0' && s[i] <= '9')

digit++;

else

other++;

i++;

}while(s[i] != '\0');

printf("%d %d %d %d %d", capital, minuscule, blank, digit, other);

}

**实验8-2-7 字符串的连接 (15 分)**

char\* str\_cat(char\* s, char\* t)

{

int n = strlen(s);

for (int i = 0; i < strlen(t); i++)

s[n++] = t[i];

s[n] = '\0';

return s;

}

**实验8-2-9 长整数转化成16进制字符串 (15 分)**

void f(long int x, char\* p)

{

if (x < 0)

sprintf(p, "-%X", -x);

else

sprintf(p, "%X", x);

}

**实验9-4 计算两个复数之积 (15 分)**

struct complex multiply(struct complex x, struct complex y)

{

struct complex result;

result.real = x.real \* y.real - x.imag \* y.imag;

result.imag = x.real \* y.imag + x.imag \* y.real;

return result;

}

**实验9-6 按等级统计学生成绩 (20 分)**

int set\_grade(struct student\* p, int n)

{

int count = 0;

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

{

if (p[i].score >= 85)

p[i].grade = 'A';

else if(p[i].score >= 70)

p[i].grade = 'B';

else if (p[i].score >= 60)

p[i].grade = 'C';

else

{

p[i].grade = 'D';

count++;

}

}

return count;

}

**实验10-2 判断满足条件的三位数 (15 分)**

int fun(int n)

{

int a, b, c;

a = n % 10;

b = n / 10 % 10;

c = n / 100;

if (a == b && a != c || a == c && a != b || b == c && b != a)

return 1;

return 0;

}

int search(int n)

{

int count = 0;

for (int i = 101; i <= n; i++)

if (fun(i))

{

int m = floor(sqrt(i) + 0.5); //floor()表示向下取整

if (m \* m == i)

count++;

}

return count;

}

**实验10-3 递归求阶乘和 (15 分)**

double fact( int n )

{

if (n <= 1)

return 1;

return n\*fact(n - 1);

}

double factsum( int n )

{

double sum = 0;

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

sum += fact(i);

return sum;

}

**实验10-4 递归实现指数函数 (15 分)**

double calc\_pow( double x, int n )

{

if (n == 0)

return 1;

return x \* calc\_pow (x, n - 1);

}

**实验10-5 递归求简单交错幂级数的部分和 (15 分)**

double calc\_pow( double x, int n )

{

if (n == 0)

return 1;

return x \* calc\_pow (x, n - 1);

}

double fn( double x, int n )

{

if (n == 1)

return x;

return calc\_pow (-1, n - 1) \* calc\_pow (x, n) + fn (x, n - 1);

}

**实验10-6 递归计算Ackermenn函数 (15 分)**

int Ack( int m, int n )

{

if (m == 0)

return n + 1;

else if (n == 0 && m > 0)

return Ack (m - 1, 1);

else if (m > 0 && n > 0)

return Ack (m - 1, Ack (m, n - 1));

}

**实验10-7 递归求Fabonacci数列 (10 分)**

int f( int n )

{

if (n == 0)

return 0;

else if (n == 1)

return 1;

return f (n - 1) + f (n - 2);

}

**实验10-8 递归计算P函数 (15 分)**

double P( int n, double x )

{

if (n == 0)

return 1;

else if (n == 1)

return x;

return ((2 \* n - 1) \* P (n - 1, x) - (n - 1) \* P (n - 2, x)) / n;

}

**实验10-9 十进制转换二进制 (15 分)**

void dectobin( int n )

{

if (n / 2 > 0)

dectobin (n / 2);

printf ("%d", n % 2);

}

**实验10-10 递归实现顺序输出整数 (15 分)**

void printdigits( int n )

{

if (n / 10 > 0)

printdigits(n / 10);

printf ("%d\n", n % 10);

}

**实验11-1-2 输出月份英文名 (15 分)**

char \*getmonth( int n ){

switch(n){

case 1: return "January";

case 2: return "February";

case 3: return "March";

case 4: return "April";

case 5: return "May";

case 6: return "June";

case 7: return "July";

case 8: return "August";

case 9: return "September";

case 10: return "October";

case 11: return "November";

case 12 :return "December";

default: return NULL;

}

}

**实验11-1-3 查找星期 (15 分)**

int getindex( char \*s )

{

char week[7][10]={"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"};

for (int i = 0; i < 7; i++)

if (strcmp (week[i], s) == 0)

return i;

return -1;

}

**实验11-1-4 计算最长的字符串长度 (15 分)**

int max\_len( char \*s[], int n )

{

int max = 0;

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

if (strlen (s[i]) > max)

max = strlen (s[i]);

return max;

}

**实验11-1-6 指定位置输出字符串 (20 分)**

char\* match(char\* s, char ch1, char ch2)

{

int i = 0, flag = 0;

char\* p;

while (s[i] != '\0' && s[i] != ch1)

i++;

p = &s[i];

while (s[i] != '\0')

{

printf("%c", s[i]);

if (s[i] == ch2)

break;

i++;

}

printf("\n");

return p;

}

**实验11-1-8 查找子串 (20 分)**

char\* search(char\* s, char\* t)

{

int i = 0, j = 0;

int flag;

char\* p;

while (s[i] != '\0')

{

if (t[j] == s[i])

{

p = &s[i];

flag = 1;

while (t[j] != '\0')

{

if (t[j] != s[i])

{

flag = 0;

break;

}

i++; j++;

}

if (flag)

return p;

}

i++; j = 0;

}

return NULL;

}

**实验11-2-1 建立学生信息链表 (20 分)**

void input()

{

struct stud\_node\* q;

q = (struct stud\_node\*)malloc(sizeof(struct stud\_node));

scanf("%d", &q->num);

while (q->num != 0)

{

scanf("%s %d", q->name, &q->score);

if (head == NULL)

{

head = q;

//head->next = NULL;

}

if (tail != NULL)

tail->next = q;

tail = q;

tail->next = NULL;

q = (struct stud\_node\*)malloc(sizeof(struct stud\_node));

scanf("%d", &q->num);

}

}

**实验11-2-2 学生成绩链表处理 (20 分)**

struct stud\_node\* createlist()

{

int num;

struct stud\_node\* head = NULL, \* tail = NULL, \* p = NULL;

//tail = (struct stud\_node\*)malloc(sizeof(struct stud\_node));

scanf("%d", &num);

while (num)

{

p = (struct stud\_node\*)malloc(sizeof(struct stud\_node));

p->num = num;

scanf("%s %d", p->name, &p->score);

p->next = NULL;

if (head == NULL)

head = p;

else

tail->next = p;

tail = p;

//tail->next = NULL;

scanf("%d", &num);

}

return head;

}

struct stud\_node\* deletelist(struct stud\_node\* head, int min\_score)

{

struct stud\_node\* ptr1, \* ptr2;

/\*要被删除的是表头结点\*/

while (head != NULL && head->score < min\_score)

{

ptr2 = head;

head = head->next;

free(ptr2);

}

if (head == NULL) //空链表

return NULL;

/\*要被删除的是非表头结点\*/

ptr1 = head;

ptr2 = head->next;//从表头的下一个结点搜索所有符合删除要求的结点

while (ptr2 != NULL)

{

if (ptr2->score < min\_score)//ptr2所指结点符合删除要求

{

ptr1->next = ptr2->next;

free(ptr2);

}

else

ptr1 = ptr2;//ptr1后移一个结点

ptr2 = ptr1->next;//ptr2指向ptr1的后一个结点

}

return head;

}

**实验11-2-3 逆序数据建立链表 (20 分)**

struct ListNode\* createlist()

{

struct ListNode\* head = NULL, \* p = NULL;

int data;

scanf("%d", &data);

while (data != -1)

{

p = (struct ListNode\*)malloc(sizeof(struct ListNode));

p->data = data;

p->next = head;

head = p;

scanf("%d", &data);

}

return head;

}

**实验11-2-4 删除单链表偶数节点 (20 分)**

struct ListNode\* createlist()

{

struct ListNode\* head = NULL, \* tail = NULL, \* p = NULL;

int data;

scanf("%d", &data);

while (data != -1)

{

p = (struct ListNode\*)malloc(sizeof(struct ListNode));

p->data = data;

p->next = NULL;

if (head == NULL)

head = p;

else

tail->next = p;

tail = p;

scanf("%d", &data);

}

return head;

}

struct ListNode\* deleteeven(struct ListNode\* head)

{

struct ListNode\* ptr1, \* ptr2;

while (head != NULL && head->data % 2 == 0)

{

ptr2 = head;

head = head->next;

free(ptr2);

}

if (head == NULL)

return NULL;

ptr1 = head;

ptr2 = head->next;

while (ptr2 != NULL)

{

if (ptr2->data % 2 == 0)

{

ptr1->next = ptr2->next;

free(ptr2);

}

else

ptr1 = ptr2;

ptr2 = ptr1->next;

}

return head;

}

**实验11-2-5 链表拼接 (20 分)**

struct ListNode\* mergelists(struct ListNode\* list1, struct ListNode\* list2)

{

struct ListNode\* head = NULL, \* tail = NULL, \* p1 = NULL, \* p2 = NULL;

int a[100], index = 0, temp;

for (p1 = list1; p1 != NULL; p1 = p1->next)

{

a[index++] = p1->data;

}

for (p1 = list2; p1 != NULL; p1 = p1->next)

{

a[index++] = p1->data;

}

for(int i=0;i<index;i++)

for (int j = i; j < index; j++)

if (a[i] > a[j])

{

temp = a[i];

a[i] = a[j];

a[j] = temp;

}

for (int i = 0; i < index; i++)

{

p2 = (struct ListNode\*)malloc(sizeof(struct ListNode));

p2->data = a[i];

p2->next = NULL;

if (head == NULL)

head = p2;

else

tail->next = p2;

tail = p2;

}

return head;

}

**实验11-2-6 奇数值结点链表 (20 分)**

struct ListNode\* readlist()

{

struct ListNode\* head = NULL, \* tail = NULL, \* p = NULL;

int data;

scanf("%d", &data);

while (data != -1)

{

p = (struct ListNode\*)malloc(sizeof(struct ListNode));

p->data = data;

p->next = NULL;

if (head == NULL)

head = p;

else

tail->next = p;

tail = p;

scanf("%d", &data);

}

return head;

}

struct ListNode\* getodd(struct ListNode\*\* L)

{

struct ListNode\* head1 = NULL, \* head2 = NULL, \* tail1 = NULL, \* tail2 = NULL, \* p;

while (\*L)

{

if ((\*L)->data % 2 == 1)

{

p = (struct ListNode\*)malloc(sizeof(struct ListNode));

p->data = (\*L)->data;

p->next = NULL;

if (head1 == NULL)

head1 = p;

else

tail1->next = p;

tail1 = p;

}

else

{

p = (struct ListNode\*)malloc(sizeof(struct ListNode));

p->data = (\*L)->data;

p->next = NULL;

if (head2 == NULL)

head2 = p;

else

tail2->next = p;

tail2 = p;

}

\*L = (\*L)->next;

}

\*L = head2;

return head1;

}

**实验11-2-7 统计专业人数 (15 分)**

int countcs(struct ListNode\* head)

{

struct ListNode\* p;

int count = 0;

for (p = head; p != NULL; p = p->next)

{

if (p->code[1] == '0')

if (p->code[2] == '2')

count++;

}

return count;

}

**实验11-2-8 单链表结点删除 (20 分)**

struct ListNode\* readlist()

{

struct ListNode\* head = NULL, \* tail = NULL, \* p = NULL;

int data;

scanf("%d", &data);

while (data != -1)

{

p = (struct ListNode\*)malloc(sizeof(struct ListNode));

p->data = data;

p->next = NULL;

if (head == NULL)

head = p;

else

tail->next = p;

tail = p;

scanf("%d", &data);

}

return head;

}

struct ListNode\* deletem(struct ListNode\* L, int m)

{

struct ListNode\* ptr1, \* ptr2, \* head = L;

while (head != NULL && head->data == m)

{

ptr2 = head;

head = head->next;

free(ptr2);

}

if (head == NULL)

return NULL;

ptr1 = head;

ptr2 = head->next;

while (ptr2 != NULL)

{

if (ptr2->data == m)

{

ptr1->next = ptr2->next;

free(ptr2);

}

else

ptr1 = ptr2;

ptr2 = ptr1->next;

}

return head;

}

**实验11-2-9 链表逆置 (20 分)**

struct ListNode\* reverse(struct ListNode\* head)

{

struct ListNode\* ptr1, \* ptr2, \* heado = NULL, \* tail;

int a[100], index = 0;

for (ptr1 = head; ptr1 != NULL; ptr1 = ptr1->next)

a[index++] = ptr1->data;

for (int i = index - 1; i >= 0; i--)

{

ptr2 = (struct ListNode\*)malloc(sizeof(struct ListNode));

ptr2->data = a[i];

ptr2->next = NULL;

if (heado == NULL)

heado = ptr2;

else

tail->next = ptr2;

tail = ptr2;

}

return heado;

}