4-1 求单链表结点的阶乘和   (15分)

本题要求实现一个函数，求单链表L结点的阶乘和。这里默认所有结点的值非负，且题目保证结果在int范围内。

**函数接口定义：**

int FactorialSum( List L );

其中单链表List的定义如下：

typedef struct Node \*PtrToNode;

struct Node {

int Data; /\* 存储结点数据 \*/

PtrToNode Next; /\* 指向下一个结点的指针 \*/

};

typedef PtrToNode List; /\* 定义单链表类型 \*/

**裁判测试程序样例：**

#include <stdio.h>

#include <stdlib.h>

typedef struct Node \*PtrToNode;

struct Node {

int Data; /\* 存储结点数据 \*/

PtrToNode Next; /\* 指向下一个结点的指针 \*/

};

typedef PtrToNode List; /\* 定义单链表类型 \*/

int FactorialSum( List L );

int main()

{

int N, i;

List L, p;

scanf("%d", &N);

L = NULL;

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

p = (List)malloc(sizeof(struct Node));

scanf("%d", &p->Data);

p->Next = L; L = p;

}

printf("%d\n", FactorialSum(L));

return 0;

}

/\* 你的代码将被嵌在这里 \*/

**输入样例：**

3

5 3 6

**输出样例：**

846

int FactorialSum( List L )

{

int sum=0,i,N=1;

while(L!=NULL)

{

N=1;

for(i=1;i<=L->Data;i++)

{

N=N\*i;

}

sum=sum+N;

L=L->Next;

}

return sum;

}

4-2 提桶寻宝   (10分)

给你准备好一个小桶，告诉你一片广袤的天地，随手一指，挖宝去吧~

说计算机话！

好吧，题目是这样的：

假设一个二维区域内藏有财宝，对于给定的此前寻到的财宝数量以及坐标(x,y)，按照以下规则在该位置寻回财宝，并更新财宝数量。

规则：

* 如果x、y均为奇数，则获得原有三分之一数量的财宝，小数部分忽略不计
* 如果x、y均为偶数，则获得 5 财宝
* 如果x、y奇偶性不同，则不获得财宝

注意：财宝数量、坐标x、坐标y都是int类型范围内的非负整数，在同一个位置可以重复寻找并获得财宝。

**函数接口定义：**

int getTreasure(int \*pBucket, int x, int y);

其中，整型参数x、y表示坐标，指针型参数pBucket指向财宝计数的整型变量，函数根据x、y和此前的财宝数目，判断新获取的财宝并修改相应数值。返回类型为int，如果获得财宝，返回1；未获得财宝，返回0。

**裁判测试程序样例：**

#include <stdio.h>

int getTreasure(int \*pBucket, int x, int y);

int main(){

int bucket,total;

bucket = total = 0;

scanf("%d",&total);

for (int i=0; i<total; i++) {

int x,y;

x = y = 0;

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

int flag = getTreasure(&bucket, x, y);

printf("[%d]:%d\n", flag, bucket);

}

}

/\*\*

你的代码将被嵌到这个位置（当然，会删除此前和此后一行的注释）

\*\*/

**输入样例：**

3

2 6

1 20

9 9

**输出样例：**

[1]:5

[0]:5

[1]:6

int getTreasure(int \*pBucket, int x, int y)

{

if(x%2==0)

{

if(y%2==0)

{

\*pBucket=\*pBucket+5;

return (1);

}

else

return (0);

}

else

{

if(y%2!=0)

{

\*pBucket=\*pBucket+\*pBucket/3;

return (1);

}

else

return (0);

}

}

4-3 递增的整数序列链表的插入   (15分)

本题要求实现一个函数，在递增的整数序列链表（带头结点）中插入一个新整数，并保持该序列的有序性。

**函数接口定义：**

List Insert( List L, ElementType X );

其中List结构定义如下：

typedef struct Node \*PtrToNode;

struct Node {

ElementType Data; /\* 存储结点数据 \*/

PtrToNode Next; /\* 指向下一个结点的指针 \*/

};

typedef PtrToNode List; /\* 定义单链表类型 \*/

L是给定的带头结点的单链表，其结点存储的数据是递增有序的；函数Insert要将X插入L，并保持该序列的有序性，返回插入后的链表头指针。

**裁判测试程序样例：**

#include <stdio.h>

#include <stdlib.h>

typedef int ElementType;

typedef struct Node \*PtrToNode;

struct Node {

ElementType Data;

PtrToNode Next;

};

typedef PtrToNode List;

List Read(); /\* 细节在此不表 \*/

void Print( List L ); /\* 细节在此不表 \*/

List Insert( List L, ElementType X );

int main()

{

List L;

ElementType X;

L = Read();

scanf("%d", &X);

L = Insert(L, X);

Print(L);

return 0;

}

/\* 你的代码将被嵌在这里 \*/

**输入样例：**

5

1 2 4 5 6

3

**输出样例：**

1 2 3 4 5 6

List Insert( List L, ElementType X )

{

List pt1,pt2,pt;

if(L->Next==NULL)

{

pt=(List)malloc(sizeof(List));

pt->Data=X;

pt->Next=NULL;

L->Next=pt;

return L;

}

pt1=L;

pt2=L->Next;

while(pt2!=NULL&&X>=pt2->Data)

{

pt1=pt2;

pt2=pt2->Next;

}

pt=(List)malloc(sizeof(List));

pt->Data=X;

pt->Next=pt2;

pt1->Next=pt;

return L;

}

4-5 函数实现字符串逆序   (15分)

本题要求实现一个字符串逆序的简单函数。

**函数接口定义：**

void f( char \*p );

函数f对p指向的字符串进行逆序操作。要求函数f中不能定义任何数组，不能调用任何字符串处理函数。

**裁判测试程序样例：**

#include <stdio.h>

#define MAXS 20

void f( char \*p );

void ReadString( char \*s ); /\* 由裁判实现，略去不表 \*/

int main()

{

char s[MAXS];

ReadString(s);

f(s);

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

return 0;

}

/\* 你的代码将被嵌在这里 \*/

**输入样例：**

Hello World!

**输出样例：**

!dlroW olleH

void f( char \*p )

{

// p指向字符串头部

char \*s=p;

// q指向字符串尾部

char \*q = s ;

while(\*q)

q++;

q -- ;

// 交换并移动指针，直到p和q交叉

while(q > s)

{

char t = \*s ;

\*s++ = \*q ;

\*q-- = t ;

}

return p ;

}

4-6 长整数转化成16进制字符串   (15分)

本题要求实现一个将长整数转化成16进制字符串的简单函数。

**函数接口定义：**

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

其中x是待转化的十进制长整数，p指向某个字符数组的首元素。函数f的功能是把转换所得的16进制字符串写入p所指向的数组。16进制的A~F为大写字母。

**裁判测试程序样例：**

#include <stdio.h>

#define MAXN 10

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

int main()

{

long int x;

char s[MAXN] = "";

scanf("%ld", &x);

f(x, s);

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

return 0;

}

/\* 你的代码将被嵌在这里 \*/

**输入样例1：**

123456789

**输出样例1：**

75BCD15

**输入样例2：**

-125

**输出样例2：**

-7D

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

{

char q;

static int i=0,j=0;

if(x<0)

{

x=-x;

\*p='-';

p++;

}

q=x%16;

if(q<10)

q=('0'+q);

else

q=('A'+q-10);

x=x/16;

j++;

i=j;

if(x>0)

f(x, p);

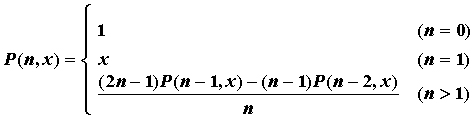
\*(p+(i-j))=q;

j--;

}

4-7 递归计算P函数   (15分)

本题要求实现下列函数P(n,x)*P*(*n*,*x*)的计算，其函数定义如下：



**函数接口定义：**

double P( int n, double x );

其中n是用户传入的非负整数，x是双精度浮点数。函数P返回P(n,x)*P*(*n*,*x*)函数的相应值。题目保证输入输出都在双精度范围内。

**裁判测试程序样例：**

#include <stdio.h>

double P( int n, double x );

int main()

{

int n;

double x;

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

printf("%.2f\n", P(n,x));

return 0;

}

/\* 你的代码将被嵌在这里 \*/

**输入样例：**

10 1.7

**输出样例：**

3.05

double P( int n, double x )

{

if(n==0)

return (1);

else if(n==1)

return (x);

else if(n>1)

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

}

4-8 统计专业人数   (15分)

本题要求实现一个函数，统计学生学号链表中专业为计算机的学生人数。链表结点定义如下：

struct ListNode {

char code[8];

struct ListNode \*next;

};

这里学生的学号共7位数字，其中第2、3位是专业编号。计算机专业的编号为02。

**函数接口定义：**

int countcs( struct ListNode \*head );

其中head是用户传入的学生学号链表的头指针；函数countcs统计并返回head链表中专业为计算机的学生人数。

**裁判测试程序样例：**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct ListNode {

char code[8];

struct ListNode \*next;

};

struct ListNode \*createlist(); /\*裁判实现，细节不表\*/

int countcs( struct ListNode \*head );

int main()

{

struct ListNode \*head;

head = createlist();

printf("%d\n", countcs(head));

return 0;

}

/\* 你的代码将被嵌在这里 \*/

**输入样例：**

1021202

2022310

8102134

1030912

3110203

4021205

#

**输出样例：**

3

int countcs( struct ListNode \*head )

{

int i=0;

while(head!=NULL)

{

if(head->code[1]=='0' && head->code[2]=='2')

i++;

head=head->next;

}

return (i);

}

4-9 输出月份英文名   (15分)

本题要求实现函数，可以返回一个给定月份的英文名称。

**函数接口定义：**

char \*getmonth( int n );

函数getmonth应返回存储了n对应的月份英文名称的字符串头指针。如果传入的参数n不是一个代表月份的数字，则返回空指针NULL。

**裁判测试程序样例：**

#include <stdio.h>

char \*getmonth( int n );

int main()

{

int n;

char \*s;

scanf("%d", &n);

s = getmonth(n);

if ( s==NULL ) printf("wrong input!\n");

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

return 0;

}

/\* 你的代码将被嵌在这里 \*/

**输入样例1：**

5

**输出样例1：**

May

**输入样例2：**

15

**输出样例2：**

wrong input!

char \*getmonth( int n )

{

char \*s[]={"January","February","March","April","May","June","July","August","September","October","November","December"};

if(n<=0||n>12)

return (NULL);

else

return (s[n-1]);

}

4-10 查找星期   (15分)

本题要求实现函数，可以根据下表查找到星期，返回对应的序号。

| 序号 | 星期 |
| --- | --- |
| 0 | Sunday |
| 1 | Monday |
| 2 | Tuesday |
| 3 | Wednesday |
| 4 | Thursday |
| 5 | Friday |
| 6 | Saturday |

**函数接口定义：**

int getindex( char \*s );

函数getindex应返回字符串s序号。如果传入的参数s不是一个代表星期的字符串，则返回-1。

**裁判测试程序样例：**

#include <stdio.h>

#include <string.h>

#define MAXS 80

int getindex( char \*s );

int main()

{

int n;

char s[MAXS];

scanf("%s", s);

n = getindex(s);

if ( n==-1 ) printf("wrong input!\n");

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

return 0;

}

/\* 你的代码将被嵌在这里 \*/

**输入样例1：**

Tuesday

**输出样例1：**

2

**输入样例2：**

today

**输出样例2：**

wrong input!

int getindex( char \*s )

{

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

int i;

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

{

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

return (i);

}

if(i==7)

return (-1);

}

4-11 计算最长的字符串长度   (15分)

本题要求实现一个函数，用于计算有n个元素的指针数组s中最长的字符串的长度。

**函数接口定义：**

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

其中n个字符串存储在s[]中，函数max\_len应返回其中最长字符串的长度。

**裁判测试程序样例：**

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#define MAXN 10

#define MAXS 20

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

int main()

{

int i, n;

char \*string[MAXN] = {NULL};

scanf("%d", &n);

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

string[i] = (char \*)malloc(sizeof(char)\*MAXS);

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

}

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

return 0;

}

/\* 你的代码将被嵌在这里 \*/

**输入样例：**

4

blue

yellow

red

green

**输出样例：**

6

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

{

int m=0,i,l;

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

{ l=strlen(s[i]);

if(m<l)

m=l;

}

return (m);

}

4-12 字符串的连接   (15分)

本题要求实现一个函数，将两个字符串连接起来。

**函数接口定义：**

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

函数str\_cat应将字符串t复制到字符串s的末端，并且返回字符串s的首地址。

**裁判测试程序样例：**

#include <stdio.h>

#include <string.h>

#define MAXS 10

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

int main()

{

char \*p;

char str1[MAXS+MAXS] = {'\0'}, str2[MAXS] = {'\0'};

scanf("%s%s", str1, str2);

p = str\_cat(str1, str2);

printf("%s\n%s\n", p, str1);

return 0;

}

/\* 你的代码将被嵌在这里 \*/

**输入样例：**

abc

def

**输出样例：**

abcdef

abcdef

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

{

char \*p;

p=strcat(s,t);

return (p);

}

4-13 使用函数判断完全平方数   (10分)

本题要求实现一个判断整数是否为完全平方数的简单函数。

**函数接口定义：**

int IsSquare( int n );

其中n是用户传入的参数，在长整型范围内。如果n是完全平方数，则函数IsSquare必须返回1，否则返回0。

**裁判测试程序样例：**

#include <stdio.h>

#include <math.h>

int IsSquare( int n );

int main()

{

int n;

scanf("%d", &n);

if ( IsSquare(n) ) printf("YES\n");

else printf("NO\n");

return 0;

}

/\* 你的代码将被嵌在这里 \*/

**输入样例1：**

10

**输出样例1：**

NO

**输入样例2：**

100

**输出样例2：**

YES

int IsSquare( int n )

{

int flag;

if((int)sqrt(n)==sqrt(n))

flag=1;

else

flag=0;

return (flag);

}

4-14 利用指针找最大值   (10分)

本题要求实现一个简单函数，找出两个数中的最大值。

**函数接口定义：**

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

其中px和px是用户传入的两个整数的指针。函数findmax应找出两个指针所指向的整数中的最大值，存放在pmax指向的位置。

**裁判测试程序样例：**

#include <stdio.h>

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

int main()

{

int max, x, y;

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

findmax( &x, &y, &max );

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

return 0;

}

/\* 你的代码将被嵌在这里 \*/

**输入样例：**

3 5

**输出样例：**

5

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

{

if(\*px<\*py)

\*pmax=\*py;

else

\*pmax=\*px;

return \*pmax;

}

4-15 十进制转换二进制   (15分)

本题要求实现一个函数，将正整数n转换为二进制后输出。

**函数接口定义：**

void dectobin( int n );

函数dectobin应在一行中打印出二进制的n。建议用递归实现。

**裁判测试程序样例：**

#include <stdio.h>

void dectobin( int n );

int main()

{

int n;

scanf("%d", &n);

dectobin(n);

return 0;

}

/\* 你的代码将被嵌在这里 \*/

**输入样例：**

10

**输出样例：**

1010

void dectobin(int n)

{

int t;

if(n==0||n==1) printf("%d",n%2);

else

{t=n;

dectobin(n=n/2);

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

}

}

4-16 分类统计字符个数   (15分)

本题要求实现一个函数，统计给定字符串中英文字母、空格或回车、数字字符和其他字符的个数。

**函数接口定义：**

void StringCount( char s[] );

其中 char s[] 是用户传入的字符串。函数StringCount须在一行内按照

letter = 英文字母个数, blank = 空格或回车个数, digit = 数字字符个数, other = 其他字符个数

的格式输出。

**裁判测试程序样例：**

#include <stdio.h>

#define MAXS 15

void StringCount( char s[] );

void ReadString( char s[] ); /\* 由裁判实现，略去不表 \*/

int main()

{

char s[MAXS];

ReadString(s);

StringCount(s);

return 0;

}

/\* Your function will be put here \*/

**输入样例：**

aZ &

09 Az

**输出样例：**

letter = 4, blank = 3, digit = 2, other = 1

void StringCount( char s[] )

{

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

int i;

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

{

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++;

}

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

}

4-17 计算两数的和与差   (10分)

本题要求实现一个计算输入的两数的和与差的简单函数。

**函数接口定义：**

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

其中op1和op2是输入的两个实数，\*psum和\*pdiff是计算得出的和与差。

**裁判测试程序样例：**

#include <stdio.h>

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

int main()

{

float a, b, sum, diff;

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

sum\_diff(a, b, &sum, &diff);

printf("The sum is %.2f\nThe diff is %.2f\n", sum, diff);

return 0;

}

/\* 你的代码将被嵌在这里 \*/

**输入样例：**

4 6

**输出样例：**

The sum is 10.00

The diff is -2.00

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

{

\*psum=op1+op2;

\*pdiff=op1-op2;

}

4-18 拆分实数的整数与小数部分   (15分)

本题要求实现一个拆分实数的整数与小数部分的简单函数。

**函数接口定义：**

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

其中x是被拆分的实数（0\le≤x<<10000），\*intpart和\*fracpart分别是将实数x拆分出来的整数部分与小数部分。

**裁判测试程序样例：**

#include <stdio.h>

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

int main()

{

float x, fracpart;

int intpart;

scanf("%f", &x);

splitfloat(x, &intpart, &fracpart);

printf("The integer part is %d\n", intpart);

printf("The fractional part is %g\n", fracpart);

return 0;

}

/\* 你的代码将被嵌在这里 \*/

**输入样例：**

2.718

**输出样例：**

The integer part is 2

The fractional part is 0.718

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

{

\*intpart=(int)x;

\*fracpart=x-(int)x;

}

4-19 求链式表的表长   (10分)

本题要求实现一个函数，求链式表的表长。

**函数接口定义：**

int Length( List L );

其中List结构定义如下：

typedef struct LNode \*PtrToLNode;

struct LNode {

ElementType Data;

PtrToLNode Next;

};

typedef PtrToLNode List;

L是给定单链表，函数Length要返回链式表的长度。

**裁判测试程序样例：**

#include <stdio.h>

#include <stdlib.h>

typedef int ElementType;

typedef struct LNode \*PtrToLNode;

struct LNode {

ElementType Data;

PtrToLNode Next;

};

typedef PtrToLNode List;

List Read(); /\* 细节在此不表 \*/

int Length( List L );

int main()

{

List L = Read();

printf("%d\n", Length(L));

return 0;

}

/\* 你的代码将被嵌在这里 \*/

**输入样例：**

1 3 4 5 2 -1

**输出样例：**

5

int Length( List L )

{

PtrToLNode node = L;

int len = 0;

if( NULL == L ){

}

while( NULL != node ){

len++;

node = node->Next;

}

return len;

}

4-20 链式表的按序号查找   (10分)

本题要求实现一个函数，找到并返回链式表的第K个元素。

**函数接口定义：**

ElementType FindKth( List L, int K );

其中List结构定义如下：

typedef struct LNode \*PtrToLNode;

struct LNode {

ElementType Data;

PtrToLNode Next;

};

typedef PtrToLNode List;

L是给定单链表，函数FindKth要返回链式表的第K个元素。如果该元素不存在，则返回ERROR。

#include <stdio.h>

#include <stdlib.h>

#define ERROR -1

typedef int ElementType;

typedef struct LNode \*PtrToLNode;

struct LNode {

ElementType Data;

PtrToLNode Next;

};

typedef PtrToLNode List;

List Read(); /\* 细节在此不表 \*/

ElementType FindKth( List L, int K );

int main()

{

int N, K;

ElementType X;

List L = Read();

scanf("%d", &N);

while ( N-- ) {

scanf("%d", &K);

X = FindKth(L, K);

if ( X!= ERROR )

printf("%d ", X);

else

printf("NA ");

}

return 0;

}

List Read() //尾插法

{

int num = 0;

scanf( "%d",&num );

List list = ( List )malloc( sizeof( struct LNode ) );

list->Next = NULL; //生成链表

list->Data = num;

List last = list; //生成指向表尾的指针

while( scanf( "%d",&num )==1&&(num!=-1)){

PtrToLNode node = ( List )malloc( sizeof( struct LNode ) );//生成新节点

node->Data = num;

node->Next = NULL;

last->Next = node; //将新节点插入表尾

last = node; //表尾指针指向新节点

}

return list;

}

ElementType FindKth( List L, int K ){

if(L==NULL) return ERROR;

while(--K){ //往后移动K-1次,查找第K个元素

if(L->Next==NULL) return ERROR; //若K大于链表的长度，返回ERROR

else L=L->Next;

}

return L->Data; //返回第K个元素

}

5-1 藏头诗   (15分)

本题要求编写一个解密藏头诗的程序。

**输入格式：**

输入为一首中文藏头诗，一共四句，每句一行。注意：一个汉字占两个字节。

**输出格式：**

取出每句的第一个汉字并连接在一起形成一个字符串并输出。同时在末尾输入一个换行符。

**输入样例：**

一叶轻舟向东流

帆稍轻握杨柳手

风纤碧波微起舞

顺水任从雅客流

**输出样例：**

一帆风顺

#include<stdio.h>

char \* ch(char s[][2000],char t[]);

int main()

{

char s[100][2000],t[1000],\*p=NULL;

int i;

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

scanf("%s",&s[i][2000]);

p=ch(s,t);

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

return 0;

}

char \* ch(char s[][2000],char t[])

{

int j;

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

{

t[2\*j]=s[j+1][0];

t[2\*j+1]=s[j+1][1];

}

t[2\*j]='\0';

return t;

}

5-2 字符串字母大小写转换   (15分)

本题要求编写程序，对一个以“#”结束的字符串，将其小写字母全部转换成大写字母，把大写字母全部转换成小写字母，其他字符不变输出。

**输入格式：**

输入为一个以“#”结束的字符串（不超过30个字符）。

**输出格式：**

在一行中输出大小写转换后的结果字符串。

**输入样例：**

Hello World! 123#

**输出样例：**

hELLO wORLD! 123

#include<stdio.h>

int main()

{

int i,k;

char s[3000];

for(i=0;i<3000;i++)

{

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

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

{ k=i;

break;

}

}

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

{

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

{ s[i]=s[i]+32; }

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

{ s[i]=s[i]-32; }

else

s[i]=s[i];

}

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

{

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

}

printf("\n");

return 0;

}

5-3 找最长的字符串   (15分)

本题要求编写程序，针对输入的N个字符串，输出其中最长的字符串。

**输入格式：**

输入第一行给出正整数N；随后N行，每行给出一个长度小于80的非空字符串，其中不会出现换行符，空格，制表符。

**输出格式：**

在一行中用以下格式输出最长的字符串：

The longest is: 最长的字符串

如果字符串的长度相同，则输出先输入的字符串。

**输入样例：**

5

li

wang

zhang

jin

xiang

**输出样例：**

The longest is: zhang

#include<stdio.h>

int main()

{

char s[1000][1000];

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

scanf("%d",&n);

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

{

for(j=0;j<80;j++)

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

if(s[i][j]=='\n')

break;

}

}

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

{

for(j=0;j<80;j++)

{

if(s[i][j]=='\n')

{

if(j>k)

{k=j;

m=i;

}

}

}

}

printf("The longest is: ");

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

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

printf("\n");

return 0;

}

5-4 通讯录的录入与显示   (10分)

通讯录中的一条记录包含下述基本信息：朋友的姓名、出生日期、性别、固定电话号码、移动电话号码。 本题要求编写程序，录入N*N*条记录，并且根据要求显示任意某条记录。

### 输入格式：

输入在第一行给出正整数N*N*（\le≤10）；随后N*N*行，每行按照格式姓名 生日 性别 固话 手机给出一条记录。其中姓名是不超过10个字符、不包含空格的非空字符串；生日按yyyy/mm/dd的格式给出年月日；性别用M表示“男”、F表示“女”；固话和手机均为不超过15位的连续数字，前面有可能出现+。

在通讯录记录输入完成后，最后一行给出正整数K*K*，并且随后给出K*K*个整数，表示要查询的记录编号（从0到N-1*N*−1顺序编号）。数字间以空格分隔。

### 输出格式：

对每一条要查询的记录编号，在一行中按照姓名 固话 手机 性别 生日的格式输出该记录。若要查询的记录不存在，则输出Not Found。

### 输入样例：

3

Chris 1984/03/10 F +86181779452 13707010007

LaoLao 1967/11/30 F 057187951100 +8618618623333

QiaoLin 1980/01/01 M 84172333 10086

2 1 7

### 输出样例：

LaoLao 057187951100 +8618618623333 F 1967/11/30

Not Found

窗体顶端

#### 题目判定

方法一

#include <stdio.h>

struct student

{

char name[11];

char birthday[11];

char sex;

char fixed[17];

char mobile[17];

};

int main()

{

struct student p[10];

int num[10];

int i, n, k;

scanf("%d", &n);

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

{

scanf("%s %s %c %s %s", p[i].name, p[i].birthday,

&p[i].sex, p[i].fixed, p[i].mobile);

}

scanf("%d", &k);

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

{

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

}

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

{

if(num[i] >= 0 && num[i] < n)

printf("%s %s %s %c %s\n", p[num[i]].name,

p[num[i]].fixed, p[num[i]].mobile, p[num[i]].sex, p[num[i]].birthday);

else

printf("Not Found\n");

}

return 0;

}

方法二

窗体底端

#include <stdio.h>

#include <string.h>

struct tong{

char name[11];

char birthday[11];

char sex;

char gnum[17];

char num[17];

};

void input();

void output();

int main()

{

int n, i;

scanf("%d", &n);

struct tong t[n];

input(t, n);

output(t, n);

return 0;

}

void input(struct tong p[], int n)

{

int i;

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

scanf("%s %s %c %s %s", p[i].name, p[i].birthday, &p[i].sex, p[i].gnum, p[i].num);

}

}

void output(struct tong q[], int n)

{

int m, j;

scanf("%d", &m);

int a[m];

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

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

}

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

if(a[j] >=0 && a[j] < n) {

printf("%s %s %s %c %s\n", q[a[j]].name, q[a[j]].gnum, q[a[j]].num, q[a[j]].sex, q[a[j]].birthday);

}

else {

printf("Not Found\n");

}

}

}

5-5 单词长度   (15分)

你的程序要读入一行文本，其中以空格分隔为若干个单词，以.结束。你要输出每个单词的长度。这里的单词与语言无关，可以包括各种符号，比如it's算一个单词，长度为4。注意，行中可能出现连续的空格；最后的.不计算在内。

**输入格式：**

输入在一行中给出一行文本，以.结束

**提示：**用scanf("%c",...);来读入一个字符，直到读到.为止。

**输出格式：**

在一行中输出这行文本对应的单词的长度，每个长度之间以空格隔开，行末没有最后的空格。

**输入样例：**

It's great to see you here.

**输出样例：**

4 5 2 3 3 4

#include<stdio.h>

int main()

{

int i,j,k=0;

char a[500];

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

{

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

if(a[i]=='.')

break;

}

for(j=0;j>=0;j++)

{

if(a[j]=='.')

{

if(k!=0)

printf("%d",k);break;

}

else if(a[j]!=' ')

k++;

else if(a[j]==' ' && a[j+1]!=' '&&k!=0&&a[j+1]!='.')

{printf("%d ",k);k=0;}

}

return 0;

}

5-6 字符串逆序   (15分)

输入一个字符串，对该字符串进行逆序，输出逆序后的字符串。

**输入格式：**

输入在一行中给出一个不超过80个字符长度的、以回车结束的非空字符串。

**输出格式：**

在一行中输出逆序后的字符串。

**输入样例：**

Hello World!

**输出样例：**

!dlroW olleH

#include <stdio.h>

#include<string.h>

int main()

{

char a[80];

int i,j,n;

gets(a);

n=strlen(a);

for(i=n-1;i>=0;i--)

{

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

if(i==0)

printf("\n");

}

return 0;

}

5-7 求简单交错序列前N项和   (15分)

本题要求编写程序,计算序列 1 - 1/4 + 1/7 - 1/10 + ... 的前N项之和。

**输入格式:**

输入在一行中给出一个正整数N。

**输出格式:**

在一行中按照“sum = S”的格式输出部分和的值S，精确到小数点后三位。题目保证计算结果不超过双精度范围。

**输入样例:**

10

**输出样例:**

sum = 0.819

#include<stdio.h>

#include<math.h>

int main()

{

int N,i;

float sum=0,S;

scanf("%d",&N);

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

{

S=(float)pow(-1,2+i)\*1/((i\*3)+1);

sum=sum+S;

}

printf("sum = %.3f\n",sum);

return 0;

}

5-8 时间换算   (15分)

本题要求编写程序，以hh:mm:ss的格式输出某给定时间再过n秒后的时间值（超过23:59:59就从0点开始计时）。

**输入格式：**

输入在第一行中以hh:mm:ss的格式给出起始时间，第二行给出整秒数n（<<60）。

**输出格式：**

输出在一行中给出hh:mm:ss格式的结果时间。

**输入样例：**

11:59:40

30

**输出样例：**

12:00:10

#include<stdio.h>

int main()

{

struct time

{

int h;

int m;

int s;

int n;

}a;

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

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

if((a.s+a.n)<59)

a.s=a.s+a.n;

else if(a.s+a.n==59)

{

a.s=0;

if(a.m==59)

{

a.m=0;

if(a.h==23)

{

a.h=0;

}

else

{

a.h=a.h+1;

}

}

else

{

a.m=a.m+1;

}

}

else if((a.s+a.n)>59)

{

a.s=(a.n+a.s)-60;

if(a.m==59)

{

a.m=0;

if(a.h==23)

{

a.h=0;

}

else

{

a.h=a.h+1;

}

}

else

{

a.m=a.m+1;

}

}

printf("%02d:%02d:%02d\n",a.h,a.m,a.s);

return 0;

}

5-9 平面向量加法   (15分)

本题要求编写程序，计算两个二维平面向量的和向量。

**输入格式:**

输入在一行中按照“x\_1*x*​1​​ y\_1*y*​1​​ x\_2*x*​2​​ y\_2*y*​2​​”的格式给出两个二维平面向量v\_1=(x\_1, y\_1)*v*​1​​=(*x*​1​​,*y*​1​​)和v\_2=(x\_2, y\_2)*v*​2​​=(*x*​2​​,*y*​2​​)的分量。

**输出格式:**

在一行中按照(x, y)的格式输出和向量，坐标输出小数点后一位（注意不能输出-0.0−0.0）。

**输入样例:**

3.5 -2.7 -13.9 8.7

**输出样例:**

(-10.4, 6.0)

#include<stdio.h>

int main()

{

double a,b,c,d,sum1,sum2;

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

sum1=a+c;

sum2=b+d;

if(sum1<0 && sum1>-0.05)

sum1 = 0.0;

if(sum2<0 && sum2>-0.05)

sum2 = 0.0;

printf("(%.1f, %.1f)\n",sum1,sum2);

return 0;

}

5-10 使用函数的选择法排序   (10分)

输入一个正整数 repeat (0<repeat<10)，做 repeat 次下列运算：

输入一个正整数 n (1<n<=10)，再输入 n 个整数存入数组 a 中，用选择法将数组 a 中的元素按升序排列，最后输出排序后的数组元素。

要求定义并调用函数 sort(a，n)，它的功能是采用选择排序算法，将数组 a 的前 n 个元素按从小到大的顺序排序，函数形参 a 的类型是整型指针，形参n的类型是int，函数的类型是void。

输出格式：数据直接以空格分隔，最后的一个数据后面没有空格。

输入输出示例：括号内为说明，无需输入输出

**输入样例:**

3 (repeat=3)

4 (n=4)

5 1 7 6

3 (n=3)

1 2 3

5 (n=5)

5 4 3 2 1

**输出样例:**

After sorted: 1 5 6 7

After sorted: 1 2 3

After sorted: 1 2 3 4 5

#include<stdio.h>

int main()

{

void sort(int \*a,int n);

int repeat,i,j,b[10],a[10][10];

scanf("%d",&repeat);

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

{

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

for(j=0;j<b[i];j++)

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

sort(a[i],b[i]);

}

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

{

printf("After sorted:");

for(j=0;j<b[i];j++)

{

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

}

printf("\n");

}

return 0;

}

void sort(int \*a,int n)

{

int i,j,k,t;

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

{k=i;

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

{

if(\*(a+k)>\*(a+j))

k=j;

}

if(i!=k)

{t=\*(a+i);

\*(a+i)=\*(a+k);

\*(a+k)=t;}

}

}

5-11 BCD解密   (10分)

BCD数是用一个字节来表达两位十进制的数，每四个比特表示一位。所以如果一个BCD数的十六进制是0x12，它表达的就是十进制的12。但是小明没学过BCD，把所有的BCD数都当作二进制数转换成十进制输出了。于是BCD的0x12被输出成了十进制的18了！

现在，你的程序要读入这个错误的十进制数，然后输出正确的十进制数。提示：你可以把18转换回0x12，然后再转换回12。

**输入格式：**

输入在一行中给出一个[0, 153]范围内的正整数，保证能转换回有效的BCD数，也就是说这个整数转换成十六进制时不会出现A-F的数字。

**输出格式：**

输出对应的十进制数。

**输入样例：**

18

**输出样例：**

12

#include <stdio.h>

int main()

{

int x;

int n, a;

scanf("%d", &x);

a = x % 16;

n = ((x / 16 )%16)\* 10 + a;

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

return 0;

}

5-12 复数四则运算   (15分)

本题要求编写程序，计算2个复数的和、差、积、商。

**输入格式：**

输入在一行中按照a1 b1 a2 b2的格式给出2个复数C1=a1+b1i和C2=a2+b2i的实部和虚部。题目保证C2不为0。

**输出格式：**

分别在4行中按照(a1+b1i) 运算符 (a2+b2i) = 结果的格式顺序输出2个复数的和、差、积、商，数字精确到小数点后1位。如果结果的实部或者虚部为0，则不输出。如果结果为0，则输出0.0。

**输入样例1：**

2 3.08 -2.04 5.06

**输出样例1：**

(2.0+3.1i) + (-2.0+5.1i) = 8.1i

(2.0+3.1i) - (-2.0+5.1i) = 4.0-2.0i

(2.0+3.1i) \* (-2.0+5.1i) = -19.7+3.8i

(2.0+3.1i) / (-2.0+5.1i) = 0.4-0.6i

**输入样例2：**

1 1 -1 -1.01

**输出样例2：**

(1.0+1.0i) + (-1.0-1.0i) = 0.0

(1.0+1.0i) - (-1.0-1.0i) = 2.0+2.0i

(1.0+1.0i) \* (-1.0-1.0i) = -2.0i

(1.0+1.0i) / (-1.0-1.0i) = -1.0

#include <stdio.h>

int main()

{

double m,n,a,b;

int i,j;

char d[4]={'+','-','\*','/'};

while(scanf("%lf %lf %lf %lf",&m,&n,&a,&b)!=EOF)

{

double c[8];

c[0]=(m+a);

c[1]=(n+b);

c[2]=(m-a);

c[3]=n-b;

c[4]=m\*a-n\*b;

c[5]=m\*b+n\*a;

c[6]=(m\*a+n\*b)/(a\*a+b\*b);

c[7]=(n\*a-m\*b)/(a\*a+b\*b);

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

{

if(c[j]>-0.05 && c[j]<0.05)

c[j]=0;

}

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

{

if(n<0)

printf("(%.1lf%.1lfi) ",m,n);

else

printf("(%.1lf+%.1lfi) ",m,n);

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

if(b<0)

printf("(%.1lf%.1lfi) ",a,b);

else

printf("(%.1lf+%.1lfi) ",a,b);

j=2\*i;

if(c[j]!=0 && c[j+1]!=0)

{

if(c[j+1]>0)

printf("= %.1lf+%.1lfi\n",c[j],c[j+1]);

else

printf("= %.1lf%.1lfi\n",c[j],c[j+1]);

}

if(c[j]!=0 && c[j+1]==0)

printf("= %.1lf\n",c[j]);

if(c[j]==0 && c[j+1]!=0)

printf("= %.1lfi\n",c[j+1]);

if(c[j]==0 && c[j+1]==0)

printf("= 0.0\n");

}

}

return 0;

}

5-13 有理数加法   (15分)

本题要求编写程序，计算两个有理数的和。

**输入格式：**

输入在一行中按照a1/b1 a2/b2的格式给出两个分数形式的有理数，其中分子和分母全是整形范围内的正整数。

**输出格式：**

在一行中按照a/b的格式输出两个有理数的和。注意必须是该有理数的最简分数形式，若分母为1，则只输出分子。

**输入样例1：**

1/3 1/6

**输出样例1：**

1/2

**输入样例2：**

4/3 2/3

**输出样例2：**

2

方法一

#include<stdio.h>

int main()

{

int a1,b1,a2,b2,k,t=1,i;

int sum1,sum2;

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

if(b1==b2)

{ sum1=a1+a2;

sum2=b1;

}

else

{

sum1=a1\*b2+a2\*b1;

sum2=b1\*b2;

if(sum1>sum2)

k=sum2;

else k=sum1;

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

{

if((sum1%i==0)&&(sum2%i==0))

{

t=i;

}

}

sum1=sum1/t;sum2=sum2/t;

}

if((sum1%sum2)==0)

printf("%d\n",sum1/sum2);

else

printf("%d/%d\n",sum1,sum2);

}

二

#include<stdio.h>

int main()

{

int a1,a2,b1,b2,k,t=1,m,w,i;

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

m=(a1\*b2)+(b1\*a2);

w=b1\*b2;

if(m>w)

k=w;

else k=m;

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

{

if((m%i==0)&&(w%i==0))

{

t=i;

}

}

m=m/t;w=w/t;

if(w==1)

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

else

printf("%d/%d\n",m,w);

return 0;

}

5-14 有理数比较   (10分)

本题要求编写程序，比较两个有理数的大小。

**输入格式：**

输入在一行中按照“a1/b1 a2/b2”的格式给出两个分数形式的有理数，其中分子和分母全是整形范围内的正整数。

**输出格式：**

在一行中按照“a1/b1 关系符 a2/b2”的格式输出两个有理数的关系。其中“>>”表示“大于”，“<<”表示“小于”，“==”表示“等于”。

**输入样例1：**

1/2 3/4

**输出样例1：**

1/2 < 3/4

**输入样例2：**

6/8 3/4

**输出样例2：**

6/8 = 3/4

#include<stdio.h>

int main()

{

int a1,a2,b1,b2;

float k,t;

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

k=(float)a1/a2;

t=(float)b1/b2;

if(k > t)

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

else if(k < t)

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

else

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

return 0;

}

5-15 计算职工工资   (15分)

给定N个职员的信息，包括姓名、基本工资、浮动工资和支出，要求编写程序顺序输出每位职员的姓名和实发工资（实发工资=基本工资+浮动工资-支出）。

**输入格式：**

输入在一行中给出正整数N。随后N行，每行给出一位职员的信息，格式为“姓名 基本工资 浮动工资 支出”，中间以空格分隔。其中“姓名”为长度小于10的不包含空白字符的非空字符串，其他输入、输出保证在单精度范围内。

**输出格式：**

按照输入顺序，每行输出一位职员的姓名和实发工资，间隔一个空格，工资保留2位小数。

**输入样例：**

3

zhao 240 400 75

qian 360 120 50

zhou 560 150 80

**输出样例：**

zhao 565.00

qian 430.00

zhou 630.00

#include <stdio.h>

int main()

{

int N,i;

float sum[1000],b[1000],c[1000],d[1000];

char a[1000][100];

scanf("%d",&N);

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

{

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

sum[i]=(b[i]+c[i])-d[i];

}

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

{

printf("%s %.2f\n",a[i],sum[i]);

}

return 0;

}

5-16 计算平均成绩   (15分)

给定N个学生的基本信息，包括学号（由5个数字组成的字符串）、姓名（长度小于10的不包含空白字符的非空字符串）和成绩（[0,100]区间内的整数），要求计算他们的平均成绩，并顺序输出平均线以下的学生名单。

**输入格式：**

输入在一行中给出正整数N（\le≤10）。随后N行，每行给出一位学生的信息，格式为“学号 姓名 成绩”，中间以空格分隔。

**输出格式：**

首先在一行中输出平均成绩，保留2位小数。然后按照输入顺序，每行输出一位平均线以下的学生的姓名和学号，间隔一个空格。

**输入样例：**

5

00001 zhang 70

00002 wang 80

00003 qian 90

10001 li 100

21987 chen 60

**输出样例：**

80.00

zhang 00001

chen 21987

#include <stdio.h>

int main()

{

int N,i,m[10],sum=0;

float ave;

char a[10][10],b[10][20];

scanf("%d",&N);

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

{

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

sum=sum+m[i];

}

ave=sum/N;

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

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

{

if(m[i]<ave)

printf("%s %s\n",b[i],a[i]);

}

return 0;

}