**复习函数这章内容，重点复习函数的调用方式、实参与形参、递归调用、静态局部变量等**

1. 若有代数式(其中*e*仅代表自然对数的底数，不是变量)，则以下能够正确表示该代数式的C语言表达式是(　　)。

　　A) sqrt(fabs(pow(n，x)＋exp(x)))

　　B) sqrt(fabs(pow(n，x)＋pow(x，e)))

　　C) sqrt(abs(n^x＋e^x))

　　D) sqrt(fabs(pow(x，n)＋exp(x)))

1. 有以下程序：

#include <stdio.h>

int f(int x)；

main()

{ int n＝1，m；

　m＝f(f(f(n)))； printf("%d\n"，m)；

}

int f(int x)

{ return x\*2； }

程序运行后的输出结果是(　　)。

　　A) 8

　　B) 2

　　C) 4

　　D) 1

1. 有以下程序：

#include <stdio.h>

fun(int x， int y)

{ static int m＝0， i＝2；

　i＋＝m＋1； m＝i＋x＋y； return m；

}

main()

{ int j＝1， m＝1， k；

　k＝fun(j，m)； printf("%d，"，k)；

　k＝fun(j，m)； printf("%d\n"，k)；

}

执行后的输出结果是(　　)。

　　A) 5， 11

　　B) 5， 5

　　C) 11， 11

　　D) 11， 5

1. 在C语言中，只有在使用时才占用内存单元的变量，其存储类型是(　　)。

　　A) auto和static

　　B) extern和register

　　C) auto和register

　　D) static和register

1. 有以下程序：

#include <stdio.h>

int fun (int x， int y )

{　if (x！＝y) return ( (x＋y) /2 )；

　 else return ( x )；

}

main()

{　int a＝4， b＝5， c＝6；

　 printf( "%d\n" ， fun(2\*a,fun(b,c)));

}

　　A) 6

　　B) 3

　　C) 8

　　D) 12

1. 有以下程序：

#include <stdio.h>

int fun()

{　 static int x＝1；

　　x\*＝2；

　　return x；

}

main()

{　 int i， s＝1；

　　for( i＝1； i<＝3； i＋＋ ) s\*＝fun()；

　　printf("%d\n"， s)；

}

程序运行后的输出结果是(　　)。

　　A) 10

　　B) 30

　　C) 0

　　D) 64

1. 在一个C源程序文件中所定义的全局变量，其作用域为(　　)。

　　A) 由具体定义位置和extern说明来决定范围

　　B) 所在程序的全部范围

　　C) 所在函数的全部范围

　　D) 所在文件的全部范围

1. 有以下程序：

#include <stdio.h>

int f( int x，int y)

{ return((y－x)\*x)； }

main()

{ int a＝3，b＝4，c＝5，d；

d＝f(f(a，b)，f(a，c))；

printf("%d\n"，d)；

}

程序运行后的输出结果是(　　)。

　　A) 7

　　B) 10

　　C) 8

　　D) 9

1. 有以下程序

#include <stdio.h>

#define N 4

void fun(int a[][N], int b[])

{ int i;

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

}

main()

{ int x[][N]={{1,2,3},{4}, {5,6,7,8},{9,10}}, y[N], i;

fun(x, y);

for (i=0;i<N; i++) printf("%d,", y[i]);

printf("\n");

}

程序的运行结果是

　　A) 1,0,7,0,

　　B) 1,2,3,4,

　　C) 1,4,5,9,

　　D) 3,4,8,10,

1. 有以下程序

#include <stdio.h>

void fun2(char a, charB)

{

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

}

char a='A', b='B';

void fun1()

{

a='C';

b='D';

}

main()

{

fun1( );

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

fun2('E', 'F');

}

程序的运行结果是

　　A) A B E F

　　B) C D E F

　　C) A B C D

　　D) C D A B

1. 若函数调用时的实参为变量时，以下关于函数形参和实参的叙述中正确的是

　　A) 函数的形参和实参分别占用不同的存储单元

　　B) 形参只是形式上的存在，不占用具体存储单元

　　C) 同名的实参和形参占同一存储单元

　　D) 函数的实参和其对应的形参共占同一存储单元

1. 以下选项中叙述错误的是

　　A) C程序函数中定义的自动变量，系统不自动赋确定的初值

　　B) 在C程序的同一函数中，各复合语句内可以定义变量，其作用域仅限本复合语句内

　　C) C程序函数中定义的赋有初值的静态变量，每调用一次函数，赋一次初值

　　D) C程序函数的形参不可以说明为static型变量

1. 以下叙述中错误的是

　　A) C程序必须由一个或一个以上的函数组成

　　B) 函数调用可以作为一个独立的语句存在

　　C) 若函数有返回值，必须通过return语句返回

　　D) 函数形参的值也可以传回给对应的实参

1. 以下程序中函数f的功能是：当flag为1时，进行由小到大排序；当flag为0时，进行由大到小排序。

#include <stdio.h>

void f(int b[], int n, int flag)

{

int i,j,t;

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

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

if (flag ? b[i]>b[j] : b[i]<b[j])

{ t=b[i]; b[i]=b[j]; b[j]=t; }

}

main()

{

int a[10]={5,4,3,2,1,6,7,8,9,10}, i;

f(&a[2], 5, 0);

f(a, 5, 1);

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

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

}

程序运行后的输出结果是

　　A) 1,2,3,4,5,6,7,8,9,10,

　　B) 3,4,5,6,7,2,1,8,9,10,

　　C) 5,4,3,2,1,6,7,8,9,10,

　　D) 10,9,8,7,6,5,4,3,2,1,

1. 有以下程序

#include <stdio.h>

int fun1( double a )

{

return (int)(a\*=a);

}

int fun2(double x, double y)

{

double a=0,b=0;

a=fun1( x );

b=fun1( y );

return (int)(a+b);

}

main()

{

double w;

w=fun2(1.1,2.0);

printf("%4.2f", w);

}

程序执行后输出结果是

　　A) 0.0

　　B) 5

　　C) 5.21

　　D) 5.00

1. 有以下程序

#include <stdio.h>

int fun( char s[])

{ int n=0;

while(\*s <= '9'&&\*s >='0') { n=10\*n+\*s-'0' ; s++; }

return (n);

}

main()

{ char s[10]={ '6', '1', '\*', '4', '\*', '9', '\*', '0', '\*'};

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

}

程序的运行结果是

　　A) 5

　　B) 9

　　C) 61

　　D) 61490

1. 有以下程序

#include <stdio.h>

int f(int n)

{ int t = 0, a=5;

if (n/2) {int a=6; t += a++; }

else {int a=7; t += a++; }

return t + a++;

}

main()

{ int s=0, i=0;

for (; i<2;i++) s += f(i);

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

}

程序运行后的输出结果是

　　A) 28

　　B) 24

　　C) 32

　　D) 36

1. 下列叙述中正确的是

　　A) C程序中所有函数之间都可以相互调用

　　B) 在C程序中main函数的位置是固定的

　　C) 在C程序的函数中不能定义另一个函数

　　D) 每个C程序文件中都必须要有一个main函数

1. 以下叙述正确的是

　　A) C语言函数可以嵌套调用，例如：fun(fun(x))

　　B) C语言程序是由过程和函数组成的

　　C) C语言函数不可以单独编译

　　D) C语言中除了main函数，其他函数不可作为单独文件形式存在

1. 有以下程序

#include <stdio.h>

void fun(int \*a, int n) /\* fun函数的功能是将a所指数组元素从大到小排序 \*/

{ int t, i, j;

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

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

if (a[i]<a[j]) { t=a[i]; a[i]=a[j]; a[j]=t; }

}

main()

{ int c[10]={1,2,3,4,5,6,7,8,9,0},i;

fun(c+4, 6);

for (i=0;i<10; i++) printf("%d,", c[i]);

printf("\n");

}

程序的运行结果是

　　A) 1,2,3,4,9,8,7,6,5,0,

　　B) 0,9,8,7,6,5,1,2,3,4,

　　C) 0,9,8,7,6,5,4,3,2,1,

　　D) 1,2,3,4,5,6,7,8,9,0,

1. 若有以下程序

#include <stdio.h>

void fun(int a[ ], int n)

{ int t,i,j;

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

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

if (a[i] > a[j]) { t=a[i]; a[i]=a[j];a[j]=t;}

}

main()

{ int c[10]={10,9,8,7,6,5,4,3,2,1},i;

fun(c, 10);

for (i=0;i<10; i++) printf("%d,", c[i]);

printf("\n");

}

则程序的输出结果是

　　A) 1,10,3,8,5,6,7,4,9,2,

　　B) 10,9,8,7,6,5,4,3,2,1,

　　C) 2,9,4,7,6,5,8,3,10,1,

　　D) 10,1,8,3,6,5,4,7,2,9,

1. 若有以下程序

#include <stdio.h>

int a=2;

int f(int m)

{ static int n;

n=0; n++; a++;

return n+m+a;

}

main()

{ int k;

for (k=0; k<4; k++) printf("%d,", f(k));

}

则程序的输出结果是

　　A) 4,5,6,7,

　　B) 4,7,10,13,

　　C) 4,6,8,10,

　　D) 3,5,7,9,

1. 若有以下程序

#include <stdio.h>

int f(int a[], int n)

{ if (n > 1)

{ int t;

t=f(a, n-1);

return t > a[n-1] ? t : a[n-1];

}

else

return a[0];

}

main()

{ int a[] = {8,2,9,1,3,6,4,7,5};

printf("%d\n", f(a, 9));

}

则程序的输出结果是

　　A) 9

　　B) 1

　　C) 8

　　D) 5

1. 有以下程序

#include <stdio.h>

fun( int a, int b )

{

int static m=0, i=2;

i=i+m+1;

m=i+a+b;

return m;

}

main()

{

int k=4, m=1, p;

p=fun( k, m);

printf("%d,",p);

p=fun( k, m);

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

}

程序运行后的输出结果是

　　A) 8,17

　　B) 7,16

　　C) 8,8

　　D) 7,17

1. 以下叙述中正确的是

　　A) 用户自己定义的函数只能调用库函数

　　B) 实用的C语言源程序总是由一个或多个函数组成

　　C) 不同函数的形式参数不能使用相同名称的标识符

　　D) 在C语言的函数内部，可以定义局部嵌套函数

1. 以下叙述中正确的是

　　A) 如果函数带有参数，就不能调用自己

　　B) 所有函数均不能接受函数名作为实参传入

　　C) 函数体中的语句不能出现对自己的调用

　　D) 函数名代表该函数的入口地址

1. 以下叙述中正确的是

　　A) 函数名允许用数字开头

　　B) 函数调用时，不必区分函数名称的大小写

　　C) 调用函数时，函数名必须与被调用的函数名完全一致

　　D) 在函数体中只能出现一次return语句

1. 有以下程序

#include <stdio.h>

int fun( int n )

{ int a;

if( n==1 ) return 1;

a = n + fun( n-1 );

return ( a );

}

main()

{ printf( "%d\n", fun( 5 ) ); }

程序的输出结果是

　　A) 9

　　B) 14

　　C) 10

　　D) 15

1. 有以下程序

#include <stdio.h>

int d=1;

void fun( int p )

{

int d=5;

d += p++;

printf("%d ", d );

}

main()

{

int a=3;

fun( a );

d += a++;

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

}

程序的输出结果是

　　A) 8 4

　　B) 9 6

　　C) 9 4

　　D) 8 5

1. 有以下程序

#include <stdio.h>

int a=1, b=2;

void fun1( int a, int b )

{ printf( "%d %d " ,a, b ); }

void fun2()

{ a=3; b=4; }

main()

{ fun1(5,6); fun2( );

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

}

程序运行后的输出结果是

　　A) 5 6 1 2

　　B) 3 4 5 6

　　C) 1 2 5 6

　　D) 5 6 3 4

1. 有以下程序

#include <stdio.h>

void func( int n )

{

static int num = 1;

num += n;

printf("%d ", num);

}

main()

{

func(3);

func(4);

printf("\n");

}

程序运行后的输出是

　　A) 4 5

　　B) 4 8

　　C) 3 5

　　D) 3 4

1. 有以下程序

#include <stdio.h>

#define N 4

void fun(int a[][N])

{ int b[N][N],i,j;

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

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

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

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

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

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

}

main()

{ int x[N][N]={ {1, 2, 3, 4},

{5, 6, 7, 8},

{9,10,11,12},

{13,14,15,16}}, i;

fun(x);

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

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

printf("\n");

}

程序的运行结果是

　　A) 13,10,7,4,

　　B) 1,6,11,16,

　　C) 4,7,10,13,

　　D) 1,5,9,13,

1. 有以下程序

#include <stdio.h>

void fun(int a[ ], int n)

{ int i,j=0,k=n-1, b[10];

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

{ b[i] =a[j];

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

j+=2; k--;

}

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

a[i] = b[i];

}

main()

{ int c[10]={10,9,8,7,6,5,4,3,2,1},i;

fun(c, 10);

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

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

printf("\n");

}

程序的运行结果是

　　A) 10,8,6,4,2,1,3,5,7,9,

　　B) 10,9,8,7,6,5,4,3,2,1,

　　C) 1,2,3,4,5,6,7,8,9,10,

　　D) 1,3,5,7,9,10,8,6,4,2,

1. 有以下程序

#include <stdio.h>

void f(int x[], int n)

{ if ( n> 1)

{ printf("%d,", x[n-1]);

f(x, n-1);

}

else

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

}

main()

{ int z[6] = {1,2,3,4,5,6};

f(z,6); printf("\n");

}

程序的运行结果是

　　A) 6,5,4,3,2,1,

　　B) 6,1,

　　C) 2,3,4,5,6,1,

　　D) 1,2,3,4,5,6,

1. 有以下程序

#include <stdio.h>

int a=2;

int f()

{ static int n=0;

int m=0;

n++; a++; m++;

return n+m+a;

}

main()

{ int k;

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

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

printf("\n");

}

程序的运行结果是

　　A) 5,7,9,

　　B) 5,6,7,

　　C) 5,8,11,

　　D) 5,5,5,

1. 以下关于C语言函数参数传递方式的叙述正确的是

　　A) 数据只能从实参单向传递给形参

　　B) 数据可以在实参和形参之间双向传递

　　C) 数据只能从形参单向传递给实参

　　D) C语言的函数参数既可以从实参单向传递给形参，也可以在实参和形参之间双向传递，可视情况选择使用

1. 有以下程序

#include <stdio.h>

#define N 4

void fun(int a[][N])

{ int b[N][N],i,j;

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

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

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

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

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

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

}

main()

{ int x[N][N]={ {1, 2, 3, 4},

{5, 6, 7, 8},

{9,10,11,12},

{13,14,15,16}

}, i;

fun(x); fun(x);

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

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

printf("\n");

}

程序的运行结果是

　　A) 16,11,6,1,

　　B) 1,6,11,16,

　　C) 4,7,10,13,

　　D) 13,10,7,4,

1. 有以下程序

#include <stdio.h>

void f(int x[], int n)

{ if ( n> 1)

{ f(&x[1], n-1);

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

}

else

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

}

main()

{ int z[6] = {1,2,3,4,5,6};

f(z,6); printf("\n");

}

程序的运行结果是

　　A) 6,5,4,3,2,1,

　　B) 1,1

　　C) 1,1,1,1,1,1,

D) 1,2,3,4,5,6,

1. 有以下程序

#include <stdio.h>

int a=2;

int f()

{ static int n;

int m;

m=n=0;

n++; a++; m++;

return n+m+a;

}

main()

{ int k;

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

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

printf("\n");

}

程序的运行结果是

　　A) 5,6,7,

　　B) 5,7,9,

　　C) 5,8,11,

　　D) 5,5,5,

1. 函数调用语句：fun( ( exp1, exp2 ), ( exp1, exp2, exp3) ); 含有的实参个数是

　　A) 1

　　B) 4

　　C) 5

　　D) 2

1. 有以下程序

#include <stdio.h>

main()

{

int i=1, j=3;

printf("%d," ,i++ );

{ int i=0;

i+=j\*2;

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

}

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

}

程序运行后的输出结果是

　　A) 1,6,3,1,3

　　B) 1,6,3,2,3

　　C) 1,6,3,6,3

　　D) 1,7,3,2,3

1. 有以下程序

#include <stdio.h>

int fun(int a,int b)

{

static int m=0, i=2;

i +=m+1; m = i + a + b;

return m;

}

main()

{

int k=4, m=1, p;

p = fun( k,m ); printf("%d,", p);

p = fun( k,m ); printf("%d\n" ,p );

}

程序运行后的输出结果是

　　A) 8,17

　　B) 8,16

　　C) 8,8

　　D) 8,20

1. 当没有指定C语言中函数形参的存储类别时，函数形参的存储类别是

　　A) 外部（extern）

　　B) 静态(static)

　　C) 寄存器(register)

　　D) 自动(auto)

1. 有如下程序

#include <stdio.h>

void convert(char ch)

{

if (ch < 'D') convert(ch+1);

printf("%c", ch);

}

main()

{

convert('A'); printf("\n");

}

程序运行后的输出结果是

　　A) ABCDDCBA

　　B) ABCD

　　C) A

　　D) DCBA

1. 有如下程序

#include <stdio.h>

int sum(int data)

{

static int init = 0;

return init += data;

}

main()

{

int i;

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

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

}

程序运行后的输出结果是

　　A) 1,3,6,10,15,

　　B) 1,2,3,4,5,

　　C) 0,0,0,0,0,

　　D) 1,1,1,1,1,

1. 以下能正确表述算式 sin(2πr+30°)的C语言表达式是

　　A) sin(2\*3.14\*r+3.14\*30/180.0)

　　B) sin(2\*π\*r+ 30 )

　　C) sin(2\*3.14\*r+ 30 )

　　D) sin(2\*3.14\*r+30\*3.14/360.0)

1. 有以下程序

#include <stdio.h>

void fun(int n)

{ int i;

if((i=n/10)!=0)

fun(i);

putchar( n%10+'0');

}

main()

{ fun(256); }

程序运行后的输出结果是

　　A) 256

　　B) 652

　　C) 2560

　　D) 52

1. 以下叙述中错误的是

　　A) 函数中的形参属于局部变量

　　B) 在函数内部定义的变量只能在本函数范围内使用

　　C) 在函数外部定义的变量在所有函数中都有效

　　D) 在不同的函数中可以使用相同名字的变量

1. 以下对C语言函数的叙述中正确的是

　　A) 调用函数时，只能把实参的值传给形参，形参的值不能传给实参

　　B) 函数既能嵌套定义，又能递归定义

　　C) 函数必须有返回值，否则不能使用

　　D) 程序中有调用关系的所有函数必须放在同一个源程序中

1. 以下针对全局变量的叙述错误的是

　　A) 全局变量的作用域是从定义位置开始至源文件结束

　　B) 全局变量是在函数外部任意位置上定义的变量

　　C) 用extern说明符可以限制全局变量的作用域

　　D) 全局变量的生存期贯穿于整个程序的运行期间

1. 以下叙述错误的是

　　A) 未经赋值的全局变量值不确定

　　B) 未经赋值的auto变量值不确定

　　C) 未经赋值的register变量值不确定

　　D) 未经赋值的静态局部变量值为0

1. 有以下程序：

#include <stdio.h>

void fun( int a， int b )

{　 int t；

　　t＝a； a＝b； b＝t；

}

main()

{　 int c[10]＝{1,2,3,4,5,6,7,8,9,0}， i；

　　for (i＝0； i<10； i＋＝2) fun(c[i]， c[i＋1])；

　　for (i＝0；i<10； i＋＋) printf("%d，"， c[i])；

　　printf("\n")；

}

程序的运行结果是(　　)。

　　A) 1,2,3,4,5,6,7,8,9,0，

　　B) 2,1,4,3,6,5,8,7,0,9，

　　C) 0,9,8,7,6,5,4,3,2,1，

　　D) 0,1,2,3,4,5,6,7,8,9，

1. 设有如下函数定义：

#include <stdio.h>

int fun( int k )

{　 if (k<1) return 0；

　　else if (k＝＝1) return 1；

　　else return fun(k－1)＋1；

}

若执行调用语句：n＝fun(3)；，则函数fun总共被调用的次数是(　　)。

　　A) 2

　　B) 3

　　C) 4

　　D) 5

1. 有以下程序

#include <stdio.h>

double f (double x);

main()

{ double a=0; int i;

for ( i=0; i<30; i+=10 ) a += f( (double)i );

printf ("%3.0f\n", a);

}

double f (double x)

{ return x\*x+1; }

程序运行后的输出结果是

　　A) 500

　　B) 401

　　C) 503

　　D) 1404

1. 有以下程序

#include <stdio.h>

void f(int b[])

{ int i;

for (i=2; i<6; i++) b[i] \*= 2;

}

main()

{ int a[10]={1,2,3,4,5,6,7,8,9,10}, i;

f(a);

for (i=0; i<10; i++) printf("%d,", a[i]);

}

程序运行后的输出结果是

　　A) 1,2,3,4,5,6,7,8,9,10,

　　B) 1,2,6,8,10,12,7,8,9,10,

　　C) 1,2,3,4,10,12,14,16,9,10,

　　D) 1,2,6,8,10,12,14,16,9,10,

/\*-------------------------------------------------------

【程序填空1】

---------------------------------------------------------

功能：已定义一个含有30个元素的数组s,函数fun1的功能是按

顺序分别赋予各元素从2开始的偶数,函数fun2则按顺序

每五个元素求一个平均值, 并将该值存放在数组w中。

-------------------------------------------------------\*/

#include <stdio.h>

long int s[30];

float w[6];

void fun1(long int s[])

{

int k,i;

for(k=2,i=0;i<30;i++)

{

/\*\*\*\*\*\*\*\*\*\*\*SPACE\*\*\*\*\*\*\*\*\*\*\*/

【?】;

k+=2;

}

}

void fun2(long int s[],float w[])

{

float sum=0.0;

int k,i;

for(k=0,i=0;i<30;i++)

{

sum+=s[i];

/\*\*\*\*\*\*\*\*\*\*\*SPACE\*\*\*\*\*\*\*\*\*\*\*/

【?】

{

w[k]=sum/5;

/\*\*\*\*\*\*\*\*\*\*\*SPACE\*\*\*\*\*\*\*\*\*\*\*/

【?】;

k++;

}

}

}

void main()

{

int i;

fun1(s);

/\*\*\*\*\*\*\*\*\*\*\*SPACE\*\*\*\*\*\*\*\*\*\*\*/

【?】;

for(i=0;i<30;i++)

{

if(i%5==0) printf("\n");

printf("%8ld",s[i]);

}

printf("\n");

for(i=0;i<6;i++)

printf("%8.2f",w[i]);

}

/\*-------------------------------------------------------

【程序填空2】

---------------------------------------------------------

功能：调用函数fun计算m=1-2+3-4+……+(n-1)-n，并输出结果。

注意：除要求填空的位置之外，请勿改动程序中其他内容。

-------------------------------------------------------\*/

#include <stdio.h>

int fun( int n)

{

int m=0,f=1,i;

/\*\*\*\*\*\*\*\*\*\*\*SPACE\*\*\*\*\*\*\*\*\*\*\*/

for(i=1;【?】;i++)

{

m+=i\*f;

/\*\*\*\*\*\*\*\*\*\*\*SPACE\*\*\*\*\*\*\*\*\*\*\*/

【?】;

}

/\*\*\*\*\*\*\*\*\*\*\*SPACE\*\*\*\*\*\*\*\*\*\*\*/

return 【?】;

}

void main()

{

printf("m=%d\n", fun(8));

printf("m=%d\n", fun(10));

}

/\*------------------------------------------------------

【程序改错1】

--------------------------------------------------------

功能：求出两个非零正整数的最大公约数，并作为函数值返回。

例如：若给num1和num2分别输入49和21，则输出的最大公约数为7。

------------------------------------------------------\*/

#include <stdio.h>

int fun(int a,int b)

{

int r,t;

if(a<b)

{

t=a;

/\*\*\*\*\*\*\*\*\*\*FOUND\*\*\*\*\*\*\*\*\*\*/

b=a;

/\*\*\*\*\*\*\*\*\*\*FOUND\*\*\*\*\*\*\*\*\*\*/

a=t;

}

r=a%b;

while(r!=0)

{

a=b;

b=r;

/\*\*\*\*\*\*\*\*\*\*FOUND\*\*\*\*\*\*\*\*\*\*/

r=a/b;

}

/\*\*\*\*\*\*\*\*\*\*FOUND\*\*\*\*\*\*\*\*\*\*/

return a;

}

main()

{

int num1,num2,a;

scanf("%d%d",&num1,&num2);

a=fun(num1,num2);

printf("the maximum common divisor is %d\n\n",a);

}

/\*------------------------------------------------------

【程序改错2】

--------------------------------------------------------

功能：实现两个字符串的连接。

例如：输入dfdfqe和12345时，则输出dfdfqe12345.

------------------------------------------------------\*/

#include <stdio.h>

main()

{

char s1[80],s2[80];

void scat(char s1[],char s2[]);

gets(s1);

gets(s2);

scat(s1,s2);

puts(s1);

}

void scat (char s1[],char s2[])

{

int i=0,j=0;

/\*\*\*\*\*\*\*\*\*\*FOUND\*\*\*\*\*\*\*\*\*\*/

while(s1[i]= ='\0')

i++;

/\*\*\*\*\*\*\*\*\*\*FOUND\*\*\*\*\*\*\*\*\*\*/

while(s2[j]= ='\0')

{

/\*\*\*\*\*\*\*\*\*\*FOUND\*\*\*\*\*\*\*\*\*\*/

s2[j]=s1[i];

i++;

j++;

}

/\*\*\*\*\*\*\*\*\*\*FOUND\*\*\*\*\*\*\*\*\*\*/

s2[j]='\0';

}

/\*------------------------------------------------

【程序设计】

--------------------------------------------------

功能：函数digit(num,k)可以实现返回整数num从右边开始

的第k位数字的值。

例如： digit(4647,3)=6

digit(23523,7)=0

------------------------------------------------\*/

#include <stdio.h>

int digit(int num,int k)

{

/\*\*\*\*\*\*\*\*\*\*Program\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*\*/

}

void main()

{

printf("digit(4647,3)=%d\n",digit(4647,3));

printf("digit(23523,7)=%d\n",digit(23523,7));

}