习题

5.1有按值传递，按地址传递，按引用传递；

按值传递不能改变实参的值，单个传递方向，按地址传递可以改变实参的值，按引用传递等于对形参的任何访问等同于对实参的访问

5.2

（1）C

A．可以只有其他函数

B.在函数中可以声明另一个函数原型

C.c程序可以由多个函数组成

D.main（）位置不固定

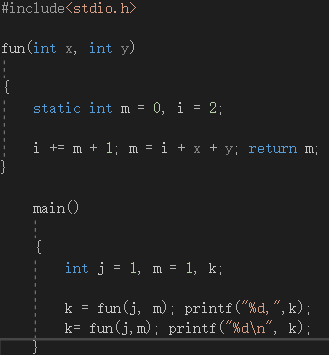
（2）B

宏定义是指在编译程序对c源程序进行预处理是用100替换标识符N

（3）C

Fun（）函数相当于求和，fun（（a++，b++，a+b），c），先执行a++得a=2，b=3，b+a=5，b+a+c=8

（4）B

fun（j，m）,i=2+0+1=3,m=3+1+1=5

fun(j,m),因为i,m是个静态局部变量，所以，m=5，i=3,i=3+5+1=9,m=9+1+1=11

（5）C

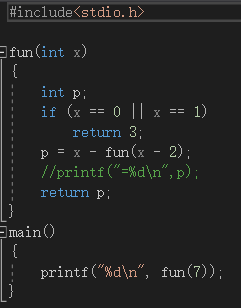
C.\*f要用括号括起来

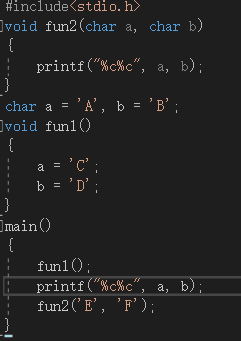
（6）B

f（3），f（2），f（1）共三次

5.3

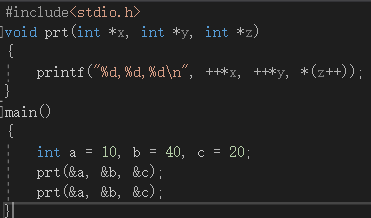
（1）

（2）

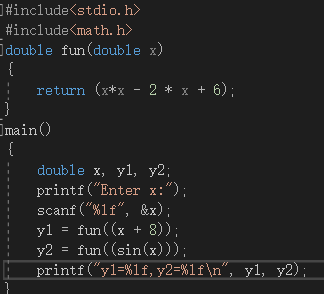


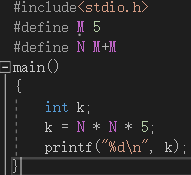


（3）

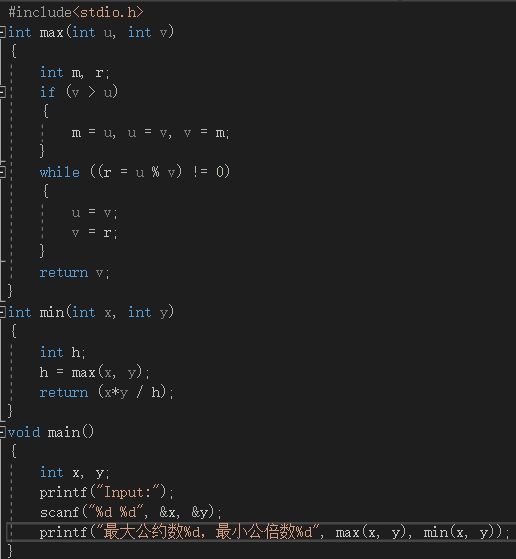
\*x按地址传递

（4）

（5）



5.4



#include<stdio.h>

int max(int u, int v)

{

int m, r;

if (v > u)

{

m = u, u = v, v = m;

}

while ((r = u % v) != 0)

{

u = v;

v = r;

}

return v;

}

int min(int x, int y)

{

int h;

h = max(x, y);

return (x\*y / h);

}

void main()

{

int x, y;

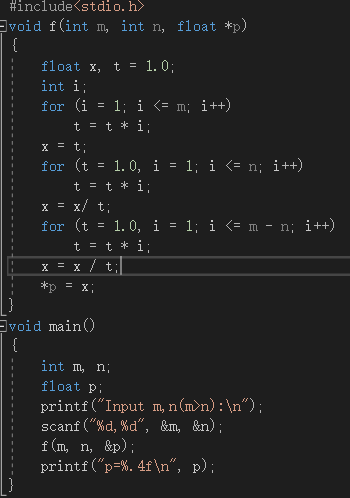
printf("Input:");

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

printf("最大公约数%d，最小公倍数%d", max(x, y), min(x, y));

}

5.5



#include<stdio.h>

void f(int m, int n, float \*p)

{

float x, t = 1.0;

int i;

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

t = t \* i;

x = t;

for (t = 1.0, i = 1; i <= n; i++)

t = t \* i;

x = x/ t;

for (t = 1.0, i = 1; i <= m - n; i++)

t = t \* i;

x = x / t;

\*p = x;

}

void main()

{

int m, n;

float p;

printf("Input m,n(m>n):\n");

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

f(m, n, &p);

printf("p=%.4f\n", p);

}

5.6

#include<stdio.h>

#include<math.h>

int huiwen(int x)

{

int y;

y = x;

int n,s=0;

while (x != 0)

{

n = x % 10;

s = s \* 10 + n;

x = x / 10;

}

if (s == y)

return 1;

else return 0;

}

int sushu(int x)

{

int sqrtn, n,i;

n = x;

sqrtn =(int)sqrt(n);

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

if (n%i == 0)

break;

if (i <= sqrtn)

return 0;

else return 1;

}

main()

{

int i;

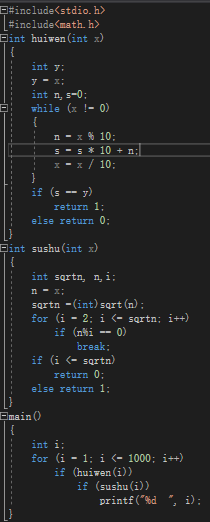
for (i = 1; i <= 1000; i++)

if (huiwen(i))

if (sushu(i))

printf("%d ", i);

}

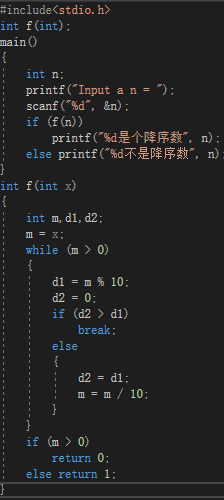


5.7

#include<stdio.h>

int f(int);

main()

{

int n;

printf("Input a n = ");

scanf("%d", &n);

if (f(n))

printf("%d是个降序数", n);

else printf("%d不是降序数", n);

}

int f(int x)

{

int m,d1,d2;

m = x;

while (m > 0)

{

d1 = m % 10;

d2 = 0;

if (d2 > d1)

break;

else

{

d2 = d1;

m = m / 10;

}

}

if (m > 0)

return 0;

else return 1;

}

5.8

#include<stdio.h>

int f(int);

main()

{

int n;

printf("Input a n = ");

scanf("%d", &n);

printf("偶数组成的数 = %d", f(n));

}

int f(int x)

{

int sum = 0, i = 1, d;

while (x > 0)

{

d = x % 10;

if (d % 2 == 0)

{

sum = sum + d\* i;

i = i \* 10;

}

x = x / 10;

}

return sum;

}



5.9

#include<stdio.h>

#include<math.h>

double arctan(double);

double f();

void main()

{

printf("PI = %lf\n",f());

}

double arctan(double x)

{

int i=1;

double b, r,xx,a; //i为分母 b为分子 xx为x的平方 a=b/i

xx = x \* x;

i = 1;

b = x;

r = 0;

while (fabs(b / i > 1e-15))

{

a = b / i;

r = (i % 4 == 1) ? r + a : r - a;

i += 2;

b = b \* xx;

}

return r;

}

double f()

{

double a, b, pi;

a = 16.0\*arctan(1 / 5.0);

b = 4.0 \* arctan(1 / 239.0);

pi = a - b;

return pi;

}



5.10

#include<stdio.h>

float fun(double h)

{

long t;

float s;

h = h \* 1000;

t = (h + 5) / 10;

s = (float)t / 100.0;

return s;

}

void main()

{

float x;

scanf("%f", &x);

printf("小数保留2位后 ，= %.3f\n", fun(x));

}



5.11

#include<stdio.h>

double f(int n,double x)

{

if (n == 0)

return 1;

else if (n == 1)

return x;

else return ((2 \* n - 1)\*x\*f(n - 1, x) - (n - 1)\*f(n - 2, x)) / (double)n;

}

int main()

{

int n;

double x;

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

printf("p(%d,%.4f) = %.6f\n", n, x, f(n, x));

}



5.12

#include<stdio.h>

#define f(x,y){int t;t = x;x = y;y = t;}

int main()

{

int a, b;

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

f(a, b);

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

}



实验6

2.（1）

当x不等于y时，返回2，反之返回x的值

在main（）函数中调用fun（）函数；

fun（b，c）输出2，fun（a，fun（b，c））输出2

（2）

float f()

{

float x1=0.0,x0;

do{

（1）x0=x1;

（2）x1=cos(x0) ；

}while（3）（fabs(x0-x1 )>=1e-6） ;

return x1;

}

void main() {

printf("结果 =%f\n",f());

}

（3）

void f(unsigned w,unsigned \*p){

unsigned s=0; //存放结果

unsigned b=1; //存放10的幂

unsigned d=0; //存放从依次取从低位到高位的数字

while(w>10) { //最高位数字不取

if(w/10)

d=w%10;

s=s+d\*b;

(1)b=b\*10;

w=w/10;

}

（2）return \*p=s;

}

void main() {

int w,v;

scanf("%d",&w);

f(w,&v);

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

}

（4）

#include<stdio.h>

int fibonacci(int n)

{

if (n < 1e-6 || n == 1)

return n;

else return fibonacci(n - 1) + fibonacci(n - 2);

}

void main()

{

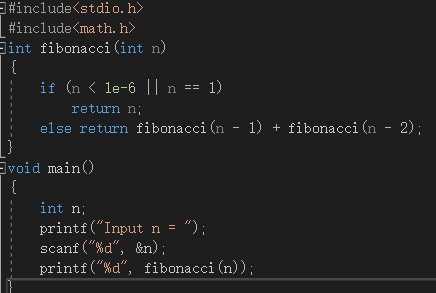
int n;

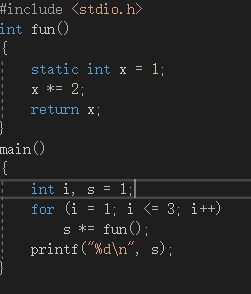
printf("Input n = ");

scanf("%d", &n);

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

}



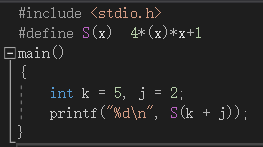
  
（5）

静态变量：空间分配和初始化在编译阶段完成，占用的空间要等到程序运行结束时才释放

存放其值的空间调用结束后不会被释放，在下次调用此函数是静态局部变量仍然保持上一次函数调用结束的值

（6）

宏定义将字符串命名为宏名字

s（k+j）被替换为4\*（k+j）\*k+j+1

s（k+j）=4\*（5+2）\*5+2+1=143