一、分析题

1. 打印结果

#include <stdio.h>

int sum(int a, int b)

{

return a + b;

}

int two(int n)

{

return n \* 2;

}

void test(int n)

{

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

}

int main()

{

test( sum(10, 11) );

return 0;

}

**/\***

**结果是42**

**我们直接看主函数,第一条语句是调用test函数,把sum(10,11)给test函数.**

**我们要先看test函数,它的作用是打印two函数的返回值,我们看从main函数到test函数的时候发生了什么:main函数把”sum(10,11)”给test函数了,test的(int n)中,n的值为”sum(10,11)”**

**在test内部调用了two函数,把”sum(10,11)”给了two函数.**

**在two函数中,two(int n)中的n依旧为”sum(10,11)”,然后返回n\*2,即返回sum(10,11)\*2.**

**我们看到sum函数,sum函数的作用是返回把两个数相加的和,即返回的是把10和11相加的和.**

**于是有:main()→test()→two()→sum()→最后sum的结果是10+11=21**

**但是我们要看看他们之间的关系:**

**sum(10,11)→21→two()→42→void test()→printf(“%d\n”,42)→main()**

**最终把42打印了出来**

**\*/**

2.

f()

{

int x = 1;

x = x +2;

printf("%d", x);

}

int main()

{

f();

f();

return 0;

}

**/\***

**结果是33**

**我们先看main函数,它直接调用了两次f函数,我们按逐语句来看**

**首先是main函数中第一次调用f():**

**在f()函数中,定义了一个x,计算后打印x,不难得知打印出3,然后返回main函数**

**然后是main函数中第二次调用f():**

**由于有作用域和生存期的存在,第一次调用的x的值已经在第一次调用完成后被释放,在第二次调用时,x被重新定义和赋值,于是又打印出3**

**因为printf总没有其他的符号,所以两个3是连在一起的.**

**\*/**

3.

void fun(int m,int n)

{

m = n;

n = m;

}

int main()

{

int x = 5, y = 7;

fun(x , y);

printf("%d,%d", x, y);

return 0;

}

**/\***

**结果是5,7**

**我们首先看主函数,定义并赋值了x,y,调用了fun函数**

**执行到fun函数的语句:**

**x的值给了m,y的值给了n,即m为5,n为7**

**到m=n这里,这条语句使得m和n的值都等于7**

**到n=m这里,一样,m==n==7**

**返回main函数**

**打印x,y的值.因为调用函数不会使得main里面定义的x,y的值发生变化,所以打印出:5,7**

**\*/**

4.

void fun(int \*m,int \*n)

{

\*m = \*n;

\*n = \*m;

}

int main()

{

int x = 5, y = 7;

fun(x , y);**/\*我觉得应该是fun(&x,&y),要不然编译会报错,我就拿fun(&x,&y)来说吧\*/**

printf("%d,%d", x, y);

return 0;

}

**/\***

**结果是7,7**

**首先看main函数,执行到fun(&x,&y)的时候,因为’&’是取址符,于是就把存储x和y地址位置给了fun函数.**

**在fun函数中(int \*m,int \*n)说明m和n是一个指向整数型变量的指针,此时指针m指向x的存储地址,指针n指向y的存储地址**

**执行到”\*m = \*n”时,表明把指针n指向的地址(也就是x)赋给指针m指向的地址(也就是y),此时变成了x的值和y的值相等,为7.**

**执行到”\*n = \*m”时,表明把指mn指向的地址(也就是y)赋给指针n指向的地址(也就是x),此时x的值和y的值还是相等,都为7.**

**返回main函数**

**因为直接对x和y的地址直接操作,所以会影响到main函数里面的x和y的值**

**打印x和y,为7,7**

**\*/**

5.

void f(int \*s)

{

s = "hello whorld";

}

int main()

{

char s[] = "hello c!";

f(s);

printf("%s", s);

return 0;

}

6.

void f(int a[])

{

int s = 0;

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

s = s + a[i];

printf("%d", s);

}

}

int main()

{

int a[5] = {1, 3, 5, 7, 9};

return 0;

}

7.

void f(int \*a, int \*b)

{

int t;

t = \*a;

\*a = \*b;

\*b = t;

}

int main()

{

int x = 3, y = 4;

f(x, y);

printf("%d,%d", x, y);

return 0;

}

8.

void f(int a[])

{

int s = 1;

for (int i = 0; i　< 5; i++) {

s = s \* a[i];

printf("%d", s);

}

}

int main()

{

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

f(x);

return 0;

}

三、编程题

1.

/\*

1.公司里面假设有2个开发人员：张三、李四

2.李四负责编写一些两个整数之间的算术运算函数，包含了：加法、减法、乘法、除法

3.张三负责编写main函数，并且要用到李四编写的算术运算函数

根据上面的情景，合理设计一套程序

1> 文件个数不限

2> 文件名自拟

3.A/B/C/D/E 5个人在某天夜里合伙去捕鱼，到凌晨时都疲惫不堪，于是各自找地方睡觉，第二天，A第一个醒来，他将鱼分成5份，把多余的一条鱼扔掉，拿走自己的一份。B第二个醒来，也将鱼分成5份，把多余的一条鱼扔掉，拿走自己的一份。C/D/E依次醒来，按同样的方法拿鱼。问他们合伙至少捕了多少条鱼？