7&3+12的值是15。请问这句话的说法是正确的吗？错误（+优先级高于&（按位与））

以下叙述中正确的是（D）

1. 结构体类型中各个成分的类型必须是一致的
2. 结构体类型中的成分只能是C语言中预先定义的基本数据类型（可以有其他已定义的结构体类型）
3. 在定义结构体类型时，编译程序就为它分配了内存空间（实例化变量才会分配内存）
4. 一个结构体类型可以由多个称为成员（或域) 的成分组成

联合体的内存大小：union{ int day;  int month;  int year;}birth; 联合体是个省内存的定义方法,里面的day month year被定义在一个4字节大小的空间中,它是day 还是month 还是year,完全取决于编程人员最后一次对它的引用.

宏定义的书写规范：

#define

P 3

#define

F(int x) {

return (P\*x\*x);}

main()

{

printf("%d\n",F(3+5));

}

宏定义换行需要有\ 宏参不需要int等数据类型 定义带宏参的宏定义要区别于函数定义不可以用return 按照正确的写法是: #define \ P 3 #define \ F(x) (P\*x\*x) main() { printf("%d\n",F(3+5)); } 3+5视为x 3\*3+5\*3+5=29

在下列选项中，合法的C语言用户标识符的是（B）

A．int B．sin C. if D. i+j

#define product(x) ((x)\*(x))

int main()

{

int i = 3, j, k;

j = product(i++);

k = product(++i);

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

return 0;

}

j = (i++\*i++);=>3\*4 第一次i++返回3 第二次i++返回4 最后i的值为5

k = (++i\*++i);=>6\*7 第三次++i返回6 第四次++i 返回7 最后i的值为7

#include <stdio.h>

int main()

{

int a=3,b=4,c=5,t=99;

if(b<a && a<c) t=a;a=c;c=t;

if(a<c && b<c) t=b;b=a;a=t;

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

}

这道题阴险，故意把if后面三条语句写在一起，其实if条件满足后，只执行紧挨着那一条

func(char para[100])

{

void \*p = malloc(100);

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

} 4 4

假定 a 为一个数组名，则下面存在错误的表达式为的（B）

A a[i] B \*a++ C \*a D \*(a+1)

这题涉及到两个知识点：

1.++的运算符优先级高于\*，这个是前提，如果没有，就不用讨论第二点了；

2.数组名a是**常量指针**，不能++；也就是说，(\*a)++则是可以的。

int main()

{

char s[]="123",\*p;

p=s;

printf("%c%c%c\n",\*p++,\*p++,\*p++);

} 结果 3 2 1（屡次采坑屡次爽）

在一个64位的操作系统中定义如下结构体：

struct st\_task

{

uint16\_t id;

uint32\_t value;

uint64\_t timestamp;

};

同时定义fool函数如下：

void fool()

{

st\_task task = {};

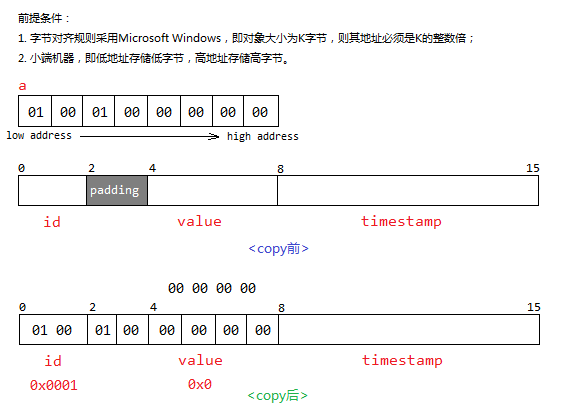
uint64\_t a = 0x00010001;

memcpy(&task, &a, sizeof(uint64\_t));

printf("%11u,%11u,%11u", task.id, task.value, task.timestamp);

}

上述fool()程序的执行结果为（）



下面两段代码中for循环分别执行了多少次？

1.

unsigned short i,j;

for(i=0, j=2; i!=j; i+=5, j+=7)

{}

2.

unsigned short i,j;

for(i=3,j=7;i!=j;i+=3,j+=7)

{}

这类问题是跑圈追赶问题：

unsigned short能表示216个数，其范围为0~216-1，j在i前2个位置，i以5的速度增长，j以7的速度增长，当增长到unsigned short表示的最大值又会返回0（以一个圈的形式循环）开始重新走直到i、j相遇，所以就是7t - 5t + 2 = 216，所以为32767次

第二个类似 (7t + 7)-(3t + 3) = 216,所以为16383次

在下列选项中，（）具有文件作用域。A 全局变量 B 静态变量（静态变量在本文件起作用，并且一经定义，任何其他改变，都会改变）

静态全局变量与普通全局变量在存储方式上完全相同

* 非静态全局变量的作用域是整个源程序中的所有文件
* 静态全局变量的作用域只是定义它的文件中