3月23日 打卡

2. Maxfd是三个套接字描述符中最大数字加上1

4.实参和与其对应的形参各占用独立的存储单元

1、形参变量只有在被调用时才分配内存单元，在调用结束时，即刻释放所分配的内存单元。因此，形参只在函数内部有效。函数调用结束返回主调用函数后则不能再使用该形参变量。

2、实参可以是常量、变量、表达式、函数等，无论实参是何种类型的量，在进行函数调用时，它们都必须有确定的值，以便把这些值传送给形参。因此应预先用赋值，输入等办法使参数获得确定值。

3、实参和形参在数量上，类型上、顺序上应严格一致，否则就会发生类型不匹配的错误。

4、在一般传值调用的机制中只能把实参传送给形参，而不能把形参的值反向地传送给实参。因此在函数调用过程中，形参值发生改变，而实参中的值不会变化。而在引用调用的机制当中是将实参引用的地址传递给了形参，所以任何发生在形参上的改变实际上也发生在实参变量上。

只要明白了值传递和地址传递，就应该能明白形参和实参的具体工作细节了。

1、值传递

    实参是变量，表达式等值。

    find（int x）{}

    y= find(z)；

   上面的例子中，**z是实参，x是形参**。**x变z不变。**

**在值传递过程中，实参和形参位于内存中两个不同地址中**，**实参先自己复制一次拷贝，再把拷贝复制给形参。**所以，在值传递过程中，形参的变化不会对实参有任何的影响。

2、地址传递（也称引用传递）

   实参是指针。

   在函数调用的时候，实参传递给你的是指针地址，地址一样也就意味着实参和形参是一样的，当你的形参发生改变时，实参也会发生改变。

    find（int  &x）{}

    y= find(z)；

    上面的例子中，**z是实参，x是形参。z随x而改变。**

3、const引用传递

    find（const int  &x）{}

    y= find(z)；

**上面的例子中，z是实参，x是形参。z不随x而改变**。

**有人就会问了，你这样做不是和值传递一样的么？NO！**

**仔细观察就会发现，在值传递中要进行两次拷贝，浪费内存资源是相当可耻的，const的出现有效避免了这种情况的出现，只需拷贝一次就够了。**

5. 格式化输出函数格式描述符



6. [C语言](https://www.baidu.com/s?wd=C%E8%AF%AD%E8%A8%80&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YLmhR3PyPhPHm4PAm4PW-W0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnH6zrH63P1b3P1ndn1fzPHfvn0)写的代码程序肯定是[源程序](https://www.baidu.com/s?wd=%E6%BA%90%E7%A8%8B%E5%BA%8F&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YLmhR3PyPhPHm4PAm4PW-W0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnH6zrH63P1b3P1ndn1fzPHfvn0) 它不能立即执行，必须经过编译成可执行代码

如果这个[源程序](https://www.baidu.com/s?wd=%E6%BA%90%E7%A8%8B%E5%BA%8F&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YLmhR3PyPhPHm4PAm4PW-W0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnH6zrH63P1b3P1ndn1fzPHfvn0)中不含有[main函数](https://www.baidu.com/s?wd=main%E5%87%BD%E6%95%B0&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YLmhR3PyPhPHm4PAm4PW-W0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnH6zrH63P1b3P1ndn1fzPHfvn0)，编译后的代码也是不可执行的 [C语言](https://www.baidu.com/s?wd=C%E8%AF%AD%E8%A8%80&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YLmhR3PyPhPHm4PAm4PW-W0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnH6zrH63P1b3P1ndn1fzPHfvn0)不是[解释执行](https://www.baidu.com/s?wd=%E8%A7%A3%E9%87%8A%E6%89%A7%E8%A1%8C&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YLmhR3PyPhPHm4PAm4PW-W0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnH6zrH63P1b3P1ndn1fzPHfvn0)的

1、【编译】是把c源程序翻译成汇编代码：\*.s； 2、【汇编】是把汇编代码翻译成二进制目标代码：\*.obj； 3、【链接】是把多个二进制目标代码文件链接成一个可执行程序；因为一个工程某个文件调用了其他c文件的函数或变量 一个程序需要上面三个步骤才能由源程序变成可执行程序。

7.



# 8.while(a<b<c)理解

首先计算a<b是否成立，再计算1<c或0<c，

如果成立则执行while循环，否则不执行

9.面向对象的三个基本特征：封装、继承、多态

编程题:

输入一个字符，用它构造一个三角形金字塔。

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

void Func(int n){

    for(int i = 0;i < 5;i++){

        for(int j = 0;j < 4 - i;j++){

            printf(" ");

        }

        for(int k = 0;k < i+1;k++){

            printf("%c ",n);

        }

        printf("\n");

    }

}

int main(){

    char n = 0;

    scanf("%c",&n);

    Func(n);

    system("pause");

    return 0;

}

3月24日 打卡

1. strlen(str)求字符串的长度，遇到\0是结束计算，不算 \0

2.



3月25日 打卡

1. 继承可以继承父类的属性和行为。封装是只提供接口并隐藏内部过程的行为。多态是指因继承产生的相关不同类，但是因为我没有多人让我继承，我也没有兄弟姐妹（就算有，他们是否富有也与我无关）。抽象是一种简化复杂现实问题的途径。简单来说就是富二代
2. fread (buffer,size,count,fp)；

buffer起点地址，size步长，count步长个数，fp FILE类型指针

fread函数的功能就是从fp所代表的文件中读取count\*size个字节的数据存放到指针buffer所指向的内存块中。

1. 无符号号整数和有符号整数相加，有符号整数转化为无符号整数，signed int b 变成了一个很大的数.

3月26日

1.



har \*str和char str[]的区别：

char \*str = "hello"; //str指向字符串常量"hello"的首地址,str存放在栈上，"hello"保存在文字常量区。通过指针只能访问字符串常量，不可以改变它

char str[] = "hello"; //数组名str也是指向字符串的首地址，str，"hello"都存放在栈上。可以通过指针去访问和修改数组内容

查看 const char\* pstring1 = "hello world"; 和const char\* pstring2 = "hello world";  的地址时，它们的地址是相同的。

1. 用数组名作为函数的实参，实际上传递的是一个指针，指向数组的首地址。
2. 引用是别名，自增等于对象自增，指针是变量，指针自增等于指向下一个地址
3. 字符串输出分情况:

①字符串长度比设定长度大就按字符串实际长度输出；

②字符串长度比设定长度小就按右对齐输出该字串,左补空格。

1. 当执行下面的程序时，如果输入ABC，则输出结果是（）.

#include  “stdio.h"

#include  “string.h"

void main()

{

    char ss[10]="1,2,3,4,5";

    gets (ss);

    strcat (ss,"6789");

    printf ("%s\n",ss);

}

第一步 ss[10] = "1,2,3,4,5";

第二步gets()之后，ss[10] = “A,B,C,\0,5"

第三步strcat()之后，ss[10] = "A,B,C,6,7,8,9"

strcat会在遇到第一个\0时开始拼接

3月27日

1. 位域的sizeof：

1) 如果相邻位域字段的类型相同，且其位宽之和小于类型的sizeof大小，则后面的字

段将紧邻前一个字段存储，直到不能容纳为止；   
2) 如果相邻位域字段的类型相同，但其位宽之和大于类型的sizeof大小，则后面的字   
段将从新的存储单元开始，其偏移量为其类型大小的整数倍；   
3) 如果相邻的位域字段的类型不同，则各编译器的具体实现有差异，VC6采取不压缩方   
式，Dev-C++采取压缩方式；   
4) 如果位域字段之间穿插着非位域字段，则不进行压缩；

5) 整个结构体的总大小为最宽基本类型成员大小的整数倍。

Struct A

{

     Char t:4;

     Char k:4;

     Unsigned short i:8;

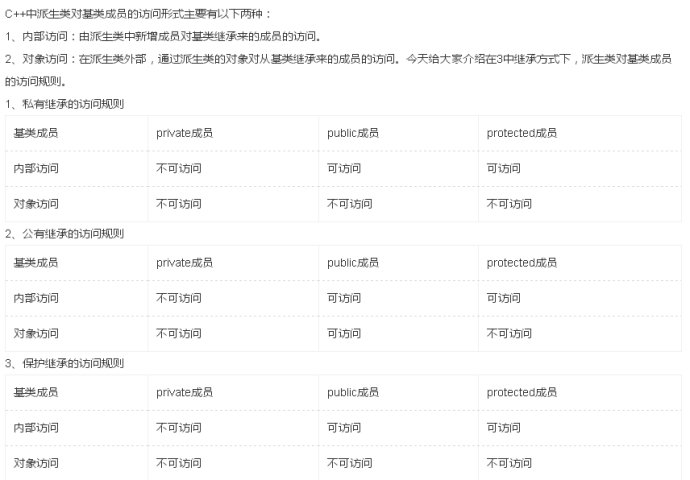
     Unsigned long m;

};

变量后面加 :   然后加数字表示位域，也就是说着代表按位来存放的，不是按字节，这是计算机为了节约空间的一种方式。char是一个字节(8个位)，所以 t和k 加起来刚好8个位，也就是一个字节。然后short 一共16个位放了8个，剩下8个不够后面long存放，所以算两个字节。因为long在32是4个字节，所以一共 1 +2 +4 = 7 。然后进行结构体对齐，所以就是8.

3月28日 打卡

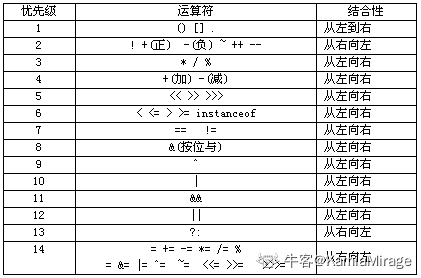
1.



.

3月29日

运算符的优先级



**运算符优先级**

括号 > 逻辑非 > 算术运算符 > 逻辑运算符

所以： !(x+y)+z-1&&y+z/2

          =!7+5-1&&4+2

          =4&&6

          =1

3月30日 打卡

1. char（1），int（4），float（4），double（8），long（8）

2.

#include <stdio.h>

int main()

{

    int i=1, j=3, k=0, sum=5;

    switch(k> -i++)

    {

        case 2: sum += 1;

        case 1: sum += j;

        case 0: sum += i; break;

        case -1: sum \*= -1;

    }

    printf("%d",sum);

    return 0;

}

1.1 - i++，指的是先使用然后再增加，比如： int i = 1; int i\_pp = i++: // i\_pp = 1;

1.2 - ++i, 指的是先加然后再使用，比如：int i = 1; int pp\_i = ++i; //pp\_i = 2;

这两点清楚后，就会知道

switch(k> -i++)  等同于 switch(k> -1) //k = 0，所以statement为true即为1。

第二个，switch case。

当switch case没有break的时候，将会按照代码行数一一执行下去。

通过第一点，得到了switch（1），所以进入case 1，由于没有break，再进入case0，但是，i，已经在switch statement的时候更新了 ，也就是说这个时候的，i，已经是2了。

所以答案为2。

3月31日 打卡

void main()

{

char \*p="abcdefgh",\*r；

long \*q；

q=(long\*)p；

q++;  //指针+1 跳过一个元素

r=(char\*)q；

printf ("%s\n",r)；

}

输出: efgh

4月1日 打卡

1. #include

void main()

{

       char\* a[ ] = { "hello", "the", "world"};

       char\*\* pa = a;

       pa++;

       cout<<\*pa<<endl;

}

分析：a是指针的数组   
char\*\* p = a; //char\*\* p = &a[0]   
p++;//p是指针自增+4，而a中元素是指针，每个正好四个字节，因此p++后恰好p= &a[1] \*p=a[1];输出"the"

2. 在C语言中,标识符的命名规则是:由字母(大、小写皆可)、数字及下划线组成，且第一个字符必须是字母或者下划线，长度为8位。在C语言中，大写字母和小写字母是有区别的，即作为不同的字母来看待，应引起注意。

4月2日 打卡

1. i++ 先输出在自加，++i 先自加在输出
2. cout一个char型指针会打印其中内容，而cout一个int型指针会打印其地址

4月3日 打卡

1. 由C语言源程序文件编译而成的目标文件的默认扩展名为obj
2. 函数可以嵌套调用但不可以嵌套定义

4月7日 打卡

1. const在定义的时候必须初始化

4月17 打卡

总结一下求补码的方法：

一、整数：

（1）正数（符号位为0）：原码，反码，补码均一致

（2）负数（符号位为1）：原码：二进制表示

                    反码：符号位不变，其余取反

                    补码：反码+1

           例如-5的原码：1000 0101 反码：1111 1010 补码：1111 1011

二、小数

（1）正数（符号位为0）：原码，反码，补码均一致

（2）负数（符号位为1）：原码：二进制表示

                                            反码：符号位不变，其余取反

                                            补码：反码+1

例：-0.5 原码：1.100 0000 反码：1.011 1111  补码：1.100 0000