指针—2.指针常量

既然有指针变量，那么就有指针常量。

常量，除了内容不可更改之外，其他操作与变量相同。

常见的指针常量有如下几类：

**一、通过取地址运算符‘&’对内存中变量取地址，得到的地址都是常量。**

例如:

1.若char ch； 则 &ch 是char \*类型。

2.若int n； 则 &n 是int \*类型。

3.若int \*p; 则 &p 是int \*\*类型。

4.若int \*\*p; 则 &p 是int \*\*\*类型。

5.若int arr[10]; 则 &arr 是 int(\*)[10]类型。

特别注意第5项，是个指向数组的指针常量。注意区别下面两个：

int \*arr[10]; arr优先和‘[]’结合，所以arr首先是个数组，是个长10的数组，数组的每个成员是指向整型的指针变量。

int (\*parr)[10]; 因为有()，所以parr优先和‘\*’结合，所以parr首先是个指针，可以指向一个长度为10，每个成员是整型的数组。

关于数组和指针的关系，下次详述。

**二、数组名都是指针常量，也包括数组常量。**

在《数组名和指针的差异中》讲过，数组名可以当做只读型的指针变量（也就是指针常量）处理，表示数组首元素的地址，因此常量数组也是一样的。例如“abcdefg”就是一个常量数组，它也可以表示数组首元素的地址，也是个指针常量，所以可以有以下的使用。

表达式 \*(“abcdefg”+2) 可以得到字符c。

因为为了书写方便，通常将 \*(p+i)写作p[i]，此处p为指针，所以上式也可以写作”abcdefg”[2]。

例：假设void UartPutByte(uint 8)可以将一字节数据输出到串口，那么完成函数void UartPutShort(uint16)将2个字节的整数按照16进制形式显示到串口，要求有前缀0x。

思路：首先输出前缀“0x”，2字节的整数共16位，每4位对应一个16进制数，转化为对应的ASCII字符输出到串口。

void UartPutShort(uint16 n)

{

signed char i;

UartPutChar ('0');

UartPutChar ('x');

for(i=3;i>=0;i--)

{

UartPutChar("0123456789ABCDEF"[（n>>(i\*4)）&0xf]);//循环将n右移并取低四位。

}

}

**三、函数名标识一段函数代码的位置，所以也可以理解为指针常量。**

在这里主要理解函数名称的含义，函数被编译之后，肯定是一段指令，当函数被加载进内存的时候，这段指令在内存中需要空间存储，而函数名则标识了内存中这段空间的首地址。

我们定义两个函数。

void fun1()

{

printf("111111111111111111\n");

}

void fun2()

{

printf("222222222222222222\n");

}

在下面代码中通过函数名相见得到两段地址的差值，然后通过fun1加差值得到fun2的地址，通过此地址调用fun2。

int main()

{

int n=(fun2-fun1); //因为函数名不能进行‘-’运算，所以转换成int型运算。

((void(\*)(void))((int)fun1+n))(); //因为函数名不能进行‘+’运算，所以转换成int型运算。运算完成之后，再转换成void(\*)(void)型的函数指针来调用函数。

return 0;

}

void(\*)(void)，因为\*首先和()结合，所以这个类型首先是个指针，然后是个指向返回值void，参数void的函数的指针。

练习：

1.“a”和‘a’的区别是什么？

2.printf(“%s\n”,”hello”+1)会输出什么？

3.char s[10]; s=”abcdef”;printf(“%s\n”,s);的结果是什么？

4.char \*s=”abcdef”; s[1]=’h’;该语句是否可以通过编译。

5.char arr[]=”abcdefg”和char \*s=”abcdefg”两个语句有什么区别？