**数组名和指针变量的差异**

数组名和指针主要有如下三点差异：

* 数组名是只读的，指针变量可读可写。
* 使用sizeof时，sizeof（数组名）得到整个数组所占字节数，sizeof（指针变量）得到指针变量所占字节数（和整型变量相同）。
* 当数组用作函数参数的时候，数组就是完完全全的指针。

例：

int \*p;

int arr[10];

对于上面两个定义。可以通过p[i]访问数组成员，也可以通过\*(arr+i)访问数组成员，这是两者相同的地方。

但是p可以重新赋值，例如p=arr是允许的，而arr=p是禁止的。同理，p++，++p都可以，但是arr不可以做自增运算，因为自增运算需要写入。

sizeof（p）为4，sizeof(arr)为40。（假设32bit机器）

当上述arr的定义出现在函数参数中时，那么在函数中，arr++和++arr，甚至arr=p都是可以的，sizeof（arr）也等于4（假设32bit机器），因为当数组作为函数参数的时候，该数组就完完全全是指针了。

下面是曾经考过的几个题目，大家可以再做做。

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

第5题（单选/3分）：

void example(char acHello[])  
{  
    printf("%d", sizeof(acHello));  
}

void main(void)  
{  
    char acHello[] = "hello";  
    example(acHello);  
}  
以上程序的输出结果是:

4  
5  
6  
hello



/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

第21题（多选/4分）：

int a[9] = {0};  
int \*p;  
p = a;  
请问以下哪些可以正确表示a[1]的值?

\*(p+1)  
\*(p++) /\*此处错位在于++i和i++的用法\*/  
\*(a++)  
\*(a+1)



/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

第19题（多选/4分）：以下为Windows NT 下的32位 C程序，  
char str[] = "Hello";  
char \*p = str;  
int n = 10;  
void \*ptr = (void \*)malloc(100);  
长度为4的是,  
  
sizeof(str)  
sizeof(p)  
sizeof(n)  
sizeof(ptr)  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

