

## 第一题

```
unsigned char *p1;
unsigned long *p2;
p1=(unsigned char *)0x801000;
p2=(unsigned long *)0x810000;
```

请问p1+5= 什么? p2+5= 什么?

答案: 801005 810014;

解析:

p1指向字符型，一次移动一个字符型，1个字节；p1+5后移5个字节，16进制表示为5；

p2指向长整型，一次移动一个长整型，4个字节，p2+5后移20字节，16进制表示为14。

{ char每次移动1个字节；short移动2个字节；int，long，float移动4个字节；double移动8个字节}

## 第二题

假设 AA 是一个类，AA\* abc () const 是该类的一个成员函数的原型。若该函数返回 this 值，当用 x.abc ()调用该成员函数后，x 的值是（）。

答案: x的值不变。

解析:

const放在函数前分为两种情况，一返回指针，此时该对象只能立即复制给新建的const char\*，而不能是char\*，意在强调值被保存在常指针中，二返回一个char也就是值，此时const无意义，应当避免。const放在函数后表明，该成员函数不得修改本对象任何成员，如果有对成员赋值语句则报错，相当于一种接口规范。

## 第三题

如果\*\*downcast\*\*是安全的(也就是，如果基类指针或者引用确实指向一个派生类对象)这个运算符会传回适当转型过的指针。如果\*\*downcast\*\*不安全，这个运算符会传回空指针(也就是说，基类指针或者引用没有指向一个派生类对象)。这个是指C++里的( )。

答案: dynamic\_cast

解析:

`dynamic_cast`将一个基类对象指针（或引用）`cast`到继承类指针，`dynamic_cast`会根据基类指针是否真正指向继承类指针来做相应处理，即会作一定的判断。对指针进行`dynamic_cast`，失败返回`null`，成功返回正常`cast`后的对象指针；对引用进行`dynamic_cast`，失败抛出一个异常，成功返回正常`cast`后的对象引用。

`reinterpret_cast`这个转换是最“不安全”的，两个没有任何关系的类指针之间转换都可以用这个转换实现。`static_cast`静态转换是最接近于C风格转换，很多时候都需要程序员自身去判断转换是否安全。`const_cast`这个转换好理解，可以将常量转成非常量。

## 第四题

下面for语句执行时的循环次数为（）。

```
int i, j;
for ( i=0, j=5;i=j; )
{ cout<<i<<j<< endl; i++; j--; }
```

答案：5

解析：

注意，中间的表达式是`i=j`，而非`i==j`，因此，只要表达式不为0就不会跳出循环。

## 第五题

以下程序的输出结果为（）。

```
main()
{int i=010,j=10;
 printf("%d,%d\n",++i,j--);
}
```

答案：9, 10

解析：

C语言中以数字1-9开头表示十进制，以0开头表示八进制，以0x开头表示十六进制。

## 第六题