

补充知识

如何理解很烦人的

`const int *p` 和 `int * const p`

指针与常量限定符

- 指针与数据组合的四种方法

1. 非常量数据的非常量指针 `type * name; // name is data, 如 int * a;`
2. 常量数据的非常量指针 `const type * name;`
3. 非常量数据的常量指针 `type * const name;`
4. 常量数据的常量指针 `const type * const name;`

- 每种组合提供不同的访问权限

指针与常量限定符

1. 非常量数据的非常量指针 `type * name`

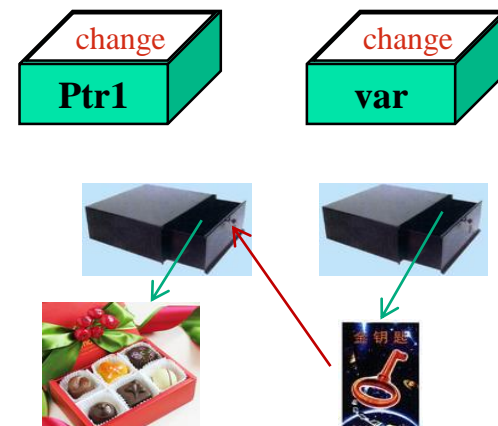
- 具有最高访问权限，指针可以修改成指向其他数据，数据也能通过指针修改，声明中不用 `const`。 (**Fig. 8.10**)
- 钥匙柜和物品柜的内容都可以换

```
...           // converting lowercase to uppercase
int main( )
{
    char phrase[] = "characters and $32.98";
    convertToUpper(phrase);
    ...
}

void convertToUpper(char *sPtr)
{
    while (*sPtr != '\0')
    {
        if (*sPtr >= 'a' && *sPtr <= 'z')
            *sPtr = toupper(*sPtr); // *sPtr -= 32;
        ++sPtr;                    // move sPtr to the next character
    }
}
```

CHARACTERS AND \$32.98

50000 Ptr1	50001 Ptr2	50002 Ptr3	50003
.....		80000 c	



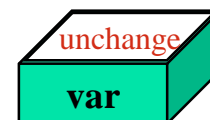
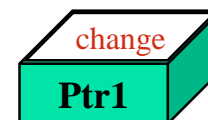
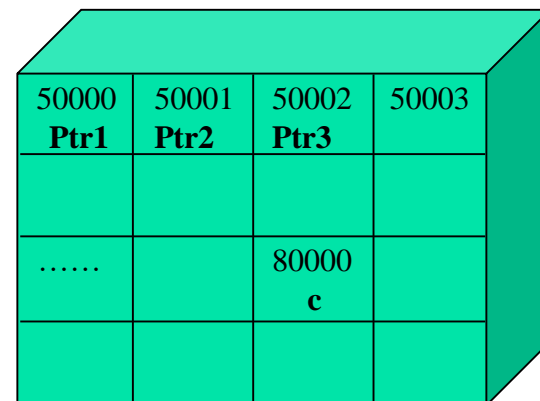
指针与常量限定符

2. 常量数据的非常量指针

const type * name

- ◆ 指针可以修改成指向其他数据，但数据不能通过指针修改。这种指针可以接收函数的数组参数，函数处理数组每个元素而不修改数据。
- ◆ 可以得到按引用调用的性能和按值调用对数据的保护。
- ◆ 钥匙柜的内容可换，物品柜的内容不可换

```
..... // (fig. 8.11)
int main( )
{
    char phrase[] = "print characters of a string";
    printCharacters (phrase);
    .....
}
void printCharacters (const char *sPtr)
{
    for( ; *sPtr != '\0'; sPtr++)
        printf("%c", *sPtr);
}
```



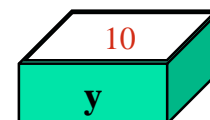
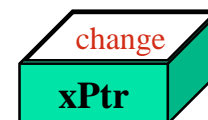
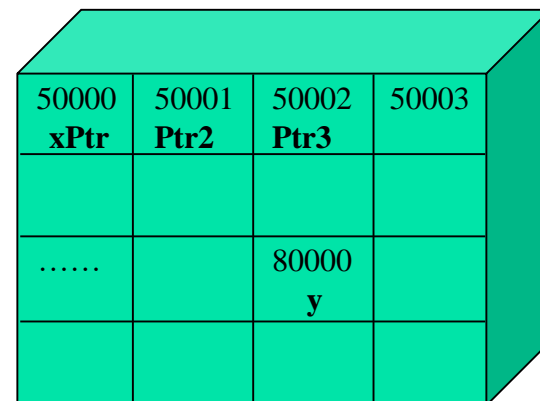
指针与常量限定符

2. 常量数据的非常量指针

`const type * name`

- ◆ 试图通过指针修改数据在编译时会产生错误。
- ◆ 可以得到按引用调用的性能和按值调用对数据的保护。
- ◆ 钥匙柜的内容可换，物品柜的内容不可换

```
..... // (fig. 8.12)
int main( )
{
    int y =10;
    f(&y);
    .....
}
void f (const int *xPtr)
{
    *xPtr = 100; // error, cannot modify a const object
}
```



指针与常量限定符

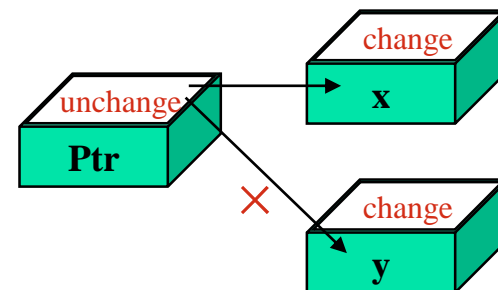
3. 非常量数据的常量指针

`type * const name`

- ◆ 指针总是指向相同的内存地址，该地址中的数据可以通过指针修改。
- ◆ 试图修改常量指针，即修改指针指向的地址会发生编译错误。
- ◆ 钥匙柜的内容不可换，物品柜的内容可换

```
.....                                     // (fig. 8.13)
int main( )
{
    int x, y ;
    int * const ptr = &x;
    *ptr = 7;
    ptr = &y;
    // error, cannot modify a const object
    .....
}
```

50000 Ptr	50001 Ptr2	50002 Ptr3	50003
.....		80000 x	



指针与常量限定符

4. 常量数据的常量指针 `const type * const name`

- ◆ 指针总是指向相同的内存地址，该内存地址的数据不能修改。具有最低访问权限。
- ◆ 钥匙柜和物品柜的内容都不可换

```
..... // (fig. 8.14)
int main( )
{
    int x = 5, y ;
    const int * const ptr = &x;
    printf("%d\n", *ptr);
    *ptr = 7; // ✗, cannot modify a const data
    ptr = &y; // ✗, cannot modify a const object
    .....
}
```

