本小节内容

C++的引用讲解
C++的布尔类型

1 C++的引用讲解

对于 C++，首先新建源文件时，名字需要叫 main.cpp,以 cpp 后缀结尾，不能像我们之前那样叫 main.c, 为什么王道数据结构都是用的 C 语言语法, 但是额外使用了 C++的引用呢, 原因是很多数据结构都采用了这个做法，比如严老师的数据结构也是这样编写的，下面我们来看一下引用的便捷性。

```
int a;
void modifynum(int &b)
{
    b=b+1;
}
调用：modifynum(a)

int *p=NULL;
void modify_pointer(int *&p)
{
    ……
    p=q;
    ……
}
调用：modify_pointer(p)
```

如上面两个例子所示，我们在修改函数外的某一变量时，**使用了引用后，在子函数内的操作和函数外操作手法一致，这样编程效率较高，对于初学者理解也非常方便，王道数据结构书籍中均采用了这种手法。**

那这种 C++的写法，和 C 语言的代码又是如何对应的呢？下面我们来看一下代码对应关系。

【例 1.1】在子函数内修改主函数的普通变量（是 C++代码，新建项目要建为 C++项目）

```
#include <stdio.h>
void modify_num(int &b)
{
    b=b+1;
}
int main() {
    int a=10;
    modify_num(a);
    printf("after modify_num a=%d\n",a);
```

```
    return 0;
}
```

上面的代码如果改为纯 C，代码如下：

【例 1.2】在子函数内修改主函数的普通变量（纯 C 代码）

```
#include <stdio.h>
void modify_num(int *b)
{
    *b=*b+1;
}
int main() {
    int a=10;
    modify_num(&a);
    printf("after modify_num a=%d\n",a);
    return 0;
}
```

【例 1.3】子函数内修改主函数的一级指针变量（这是是重要的！）

```
#include <stdio.h>
void modify_pointer(int* &p,int *q)
{
    p=q;
}
int main() {
    int *p=NULL;
    int i=10;
    int *q=&i;
    modify_pointer(p,q);
    printf("after modify_pointer *p=%d\n",*p);
    return 0;
}
```

上面的代码如果改为纯 C，就需要使用到二级指针，二级指针我们没有讲解，因为对于考研初试是使用不到的，因此下面的代码不明白完全没关系，代码如下：

```
#include <stdio.h>
void modify_pointer(int **p,int *q)//相对于 C++这里是 int **p;
{
    *p=q;//这里的写法和例 1.2 中的是非常类似的
```

```
}
int main() {
    int *p=NULL;
    int i=10;
    int *q=&i;
    modify_pointer(&p,q);//相对于 C++这里是&p
    printf("after modify_pointer *p=%d\n",*p);
    return 0;
}
```

2 C++的布尔类型

布尔类型在 C 语言没有，是 C++的，有 true 和 false，通过下面代码我们来理解一下它们：

【例 2.1】 布尔类型也是有值的

```
#include <stdio.h>

//设置布尔值的好处是提升了代码的可阅读性
int main() {
    bool a=true;
    bool b= false;
    printf("a=%d,b=%d\n", a,b);
    return 0;
}
```