**上机作业二**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 姓名: | yyh | 学号： |  | 日期： | 2018.09.11 |

|  |
| --- |
| **【实验目的】**   1. 函数的编写和调用，掌握值传递、引用传递和指针传递 2. 内联函数应用 3. 函数缺省值和函数重载 |
| **【实验总结】**  请在以下总结实验中发现的问题和解决办法或心得体会。请勿黏贴过多源码。  问题1：引用的使用，对数组的引用不能引用整个数组。  解决：因为数组头本质是一个指针（存有首个元素的地址），而指针是不能引用的。  只能引用数组元素。  问题2：第四个作业中加密出现乱码。  解决：实际上这些乱码仍然是字符变量，但它们是不可显示的字符（具体参见心得体会3）。它们并不影响对原字符串的加密，对它们仍然可以进行解密操作（部分情况可能不行，见心得体会4），它们解密后仍然可以得到原字符串。  心得体会3：对于第4个作业，异或运算’^’是让变量按照二进制编码对应这每一位进行异或运算。可以肯定的是两字符进行异或运算后仍然是一个字符，但是两个正常字符加密后进行打印可能出现乱码、空字符、换行等，例如的结果打印出来就是空字符。这是由于异或后的字符可能是控制字符，是不可显示的字符（或者不能直观看见的字符），例如，已知而通过查询ASCII码表知道表示的是水平制表符，是不可显示的字符。  Figure 部分ASCII码表  心得体会4：对于第4个作业，通过异或运算可以得到任意满足。而正好对应了字符串结束符’\0’，在输出字符串结束符时就会结束整个字符串的输出（貌似string类型的字符串不受此影响，但用char[]字符数组来定义字符串就会受影响），无论后面是否还有字符，这样一来，就会发现输出的密文比原文少一些字符。  心得体会5：对于第4个作业的破译钥匙，可以结合异或运算的性质进行破译：使用表示key的第i位，使用表示字符‘A’，用表示字符‘b’，用表示字符‘#’,表示异或运算，可以得到：  将（1），（2），（3）异或后结合（4）得：  这样就得到了key。  总结起来就是把字符串”AAAAAAAA”输入得到c1,将字符串“bbbbbbbb”输入得到c2,将字符串“########”输入得到c3,再将c1 c2 c3异或就可以得到key了。  可以参考home\_work\_4.2.cpp设计破译程序。  心得体会6：关于内联函数。内联函数的目的在于减少对函数的调用，从而减少每次调用函数带来的额外时间。虽然内联函数同宏定义类似均可以被自己的函数内容代替，但是内联函数的出现大大增加了程序的可读性和易改性。内联函数与宏定义相比最大的优势在于它能对变量进行检查。内联函数适用于简单的函数表达。内联函数不宜出现多次，否则增加不必要的编译时间。    心得体会7：可以使用codeblocks看汇编代码。在汇编代码下，int &a与int \*a的代码完全一致。参考<http://www.nowamagic.net/librarys/veda/detail/2335>。  心得体会8：函数的缺省值，注意是顺序先后再前，在第一次函数定义/函数声明时定义缺省值，要注意其结合重载后具有的二义性。 |