一、实验目的：

AES 算法的实现

二、实验环境：

运行环境：Xcode version 5.1.1

系统环境：Mac OS X 10.10

三、编写语言：

C++

四、实验内容：

该程序主要用于实现数据的 AES 加密及解密，具体内容如下：

1. 程序能够读入指定的 txt 密钥文档；
2. 可以对指定的 txt 文档进行加密或者脱密处理，并生成对应的密文或者明文 txt 文档。

五、实验步骤：

AES 算法在整体结构上采用的是 Square 结构而不是 Feistel 结构，该结构由 4 个不同的阶段组成，包括 1 个混乱和 3 个代换。

①：字节代换(SubBytes)，用一个 S 盒完成分组中的按字节的代换。

②：行移位代换(ShiftRows)，一个简单的置换。

③：列混淆(MixColumns)，一个利用在域 GF(2^8)上的算术特征的代换。

④：轮密钥加(AddRoundKey)，利用当前分组和扩展密钥的一部分进行按位异或(XOR)。

在密钥方面，包括两个部分：密钥扩展和轮密钥选取.

密钥扩展：密钥 bit 的总数＝分组长度×（轮数 Round＋1）例如当分组长度为 128bits 和轮数 Round 为 10 时，轮密钥长度为 128×(10＋1)＝1408bits。

将密码密钥扩展成一个扩展密钥。

轮密钥选取：第一个轮密钥由扩展密钥的第一个 Nb 个 4 字节字，第二个圈密钥由接下来的 Nb 个 4 字节字组成，以此类推。

上述的实验步骤每个步骤的划分都明确，分模块实现这些功能即可，函数原型如下：

```
void SubBytes(unsigned char matrix[][4]);
void ShiftRows(unsigned char matrix[][4]);
void MixColumns(unsigned char matrix[][4]);
void AddRoundKey(unsigned char matrix[][4],int round);
void KeyExpansion(unsigned char key[16]);
void InvSubBytes(unsigned char matrix[][4]);
void InvShiftRows(unsigned char matrix[][4]);
void InvMixColumns(unsigned char matrix[][4]);
```

六、实验的难点

1）首先，在 S 盒功能的实现上，一开始考虑自己写代码模拟 S 盒的运算过程，但后来发现还不如直接建表访问来的方便，并且也能加快运行速度，于是 S 盒和逆 S 盒都采用了建表访问的方式

2）在列混合变换中，涉及到在域 GF(2^8)上的算术特征的代换，这部分的数学知识比较欠缺，看了一些定义和推导之后还是有些云里雾里，就依样画葫芦的逐

步进行操作（部分相乘的矩阵并不是特别了解其功能）
3）文件读写上采用的是 string 形式的文件读写，加密过程时，首先从文件中读取字符，存在类型是 unsigned char 的数组中（其实存放的是对应的 ASCII 码值），然后将 ASCII 值传入 AES 算法的类函数中进行加密的运算，如果直接输出该数组，那么输出的是一系列的乱码（虽然在程序上并没有错误），但为了密文的可读性以及教师核收的方便，以 16 进制的形式输出 ASCII 码，而不是直接输出 ASCII 码对应的字符。另外在脱密过程中，由于需要从文件中读取两位的十六进制（如 3,C），再将其转化为对应的 ASCII 码（3C），这一过程无法直接通过文件实现一步操作，只能够先将分割的 3 和 C 合并起来（change 函数），再进一步进行处理，虽然逻辑没问题，但在处理过程上显得有些绕弯。

七、程序运行结果展示：

通过网上的在线 AES 加脱密模拟系统，有下列一组测试数据
密钥：1234567890123456
明文：testtesttesttest
密文：4c7e0db9c5a9d2ef98d1940e017608f2
测试如下：

加密过程:

起初在 input.txt 中放入 testtesttesttest，key.txt 中放入 1234567890123456，然后运行如下：（注：input.txt 和 key.txt 都放在 debug 文件夹下故无需包含路径）

```
请输入密钥文件位置及文件名
key.txt
加密操作输入1  解密操作输入2
1
请输入明文位置及文件名
input.txt
请输入存放密文的位置及文件名
output.txt
Program ended with exit code: 0
```

output.txt 的内容则为：4c7e0db9c5a9d2ef98d1940e017608f2

解密过程：

起初在 input.txt 中放入 4c7e0db9c5a9d2ef98d1940e017608f2，key.txt 中放入 1234567890123456，然后运行如下：（注：input.txt 和 key.txt 都放在 debug 文件夹下故无需包含路径）

```
请输入密钥文件位置及文件名
key.txt
加密操作输入1  解密操作输入2
2
请输入密文位置及文件名
input.txt
请输入存放明文的位置及文件名
output.txt
Program ended with exit code: 0
```

output.txt 的内容则为：testtesttesttest （完结）