第 1 关：基本测试

根据 S-DES 算法编写和调试程序，提供 GUI 解密支持用户交互。输入可以是 8bit 的数据和 10bit 的密钥，输出是 8bit 的密文。加解密的测试样例如下：

加密测试样例：

明文	密钥	密文
11111111	1111111111	1100100111001001110010011100100111001001110010011100100111001001
11111111	0000000000	0010110000101100001011000010110000101100001011000010110000101100
10101010	1100011000	1100100110011100110010011001110011001001100111001100100110011100
11101011	0110110110	0010110000101100001011001101100100101100110110010010110000101100

加密界面：

解密测试样例：

密文	密钥	明文
1100100111001001110010011100100111001001110010011100100111001001	1111111111	11111111
0010110000101100001011000010110000101100001011000010110000101100	0000000000	11111111
1100100110011100110010011001110011001100100110011100110010011100	1100011000	10101010
0010110000101100001011001101100100101100110110010010110000101100	0110110110	11101011

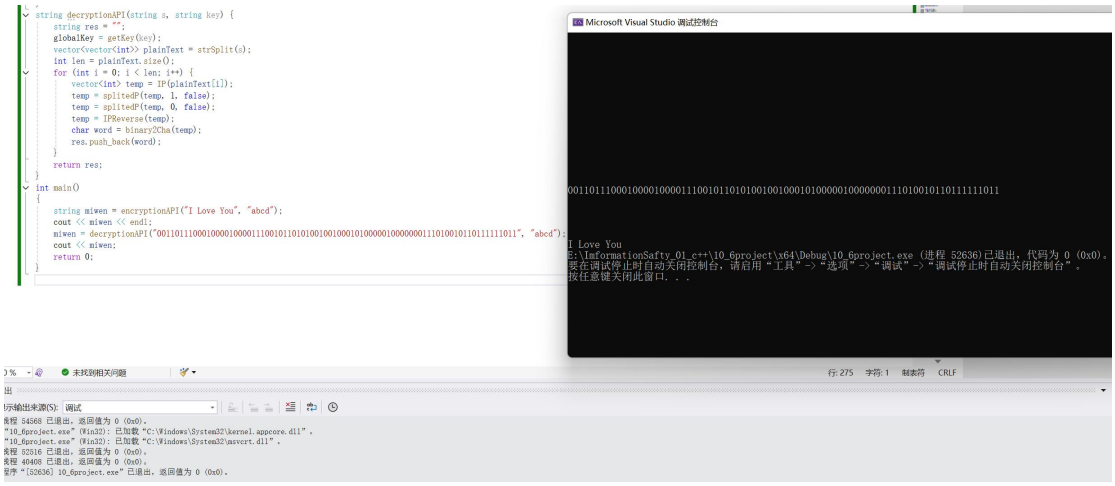
解密界面：

第 2 关：交叉测试

考虑到是**算法标准**，所有人在编写程序的时候需要使用相同算法流程和转换单元(P-Box、S-Box 等)，以保证算法和程序在异构的系统或平台上都可以正常运行。

设有 A 和 B 两组位同学(选择相同的密钥 K)；则 A、B 组同学编写的程序对明文 P 进行加密得到相同的密文 C；或者 B 组同学接收到 A 组程序加密的密文 C，使用 B 组程序进行解密可得到与 A 相同的 P。

A 同学的结果如下：



B 同学的结果如下：

综上，A、B 同学加解密结果相同，该算法和程序可以在不同的平台上正常运行

第 3 关：扩展功能

考虑到向实用性扩展，加密算法的数据输入可以是 ASII 编码字符串(分组为 1 Byte)，对应地输出也可以是 ACII 字符串(很可能是乱码)。

算法说明：因为字符加密以后，可能出现非可见字符的 ASCII 码，所以会出现多处空白，这将导致密文不能加密回去。所以，我们将加密出来的密文设置为 0-1 字符串的形式，而不是字符的形式。

第 4 关：暴力破解

假设你找到了使用相同密钥的明、密文对(一个或多个)，请尝试使用暴力破解的方法找到正确的密钥 Key。在编写程序时，你也可以考虑使用多线程的方式提升破解的效率。请设定时间戳，用视频或动图展示你在多长时间内完成了暴力破解。

```
string plainText = generateRandomString(10);//生成随机明文
string miwen = encryptionAPI(plainText, "abcd");//通过加密获得密文
cout << miwen << endl;//展示密文
int count = 1<<10;//10位密钥密钥空间为1024。
cout << count << endl;
string mingwen;
clock_t start = clock();//记录算法开始时间点
for (int i = 0; i < count; i++) {
    vector<int> forceKey = forceInt2Binary(i);//尝试值为i的key
    mingwen = forceDecryptionAPI(miwen, forceKey);//通过暴力key得到明文
    if (mingwen == plainText) {//如果解密出的明文与本来的明文相等则枚举key成功
        cout << "Success" << endl;
        clock_t end = clock();
        printf("算法运行了%d ms", end - start);//得到暴力破解所需时间
        break;
    }
}
```

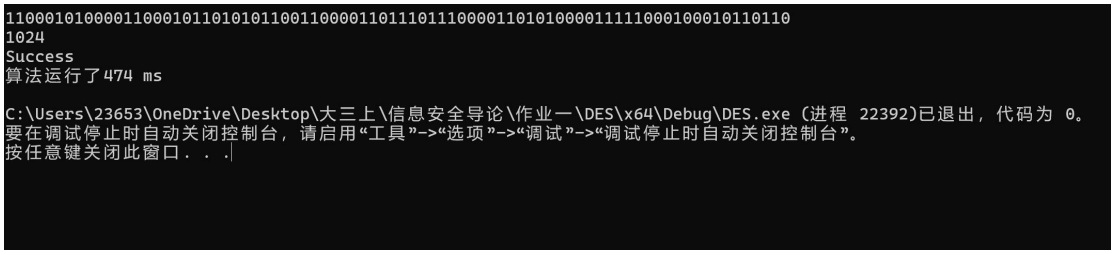
```
0100111100001110100010000111110100101111010100101001101100111011001110000111100000
1024
Success
算法运行了475 ms
C:\Users\23653\OneDrive\Desktop\大三上\信息安全导论\作业一\DES\x64\Debug\DES.exe (进程 20308)已退出，代码为 0。
要在调试停止时自动关闭控制台，请启用“工具”->“选项”->“调试”->“调试停止时自动关闭控制台”。
按任意键关闭此窗口. . .
```

暴力破解成功，用时只需百毫秒左右。

第 5 关：封闭测试

根据第 4 关的结果，进一步分析，对于你随机选择的一个明密文对，是不是有不止一个密钥 Key？进一步扩展，对应明文空间任意给定的明文分组 P_n，是否会出现选择不同的密钥$K_i \neq K_j$加密得到相同密文C_n的情况？

```
string plainText = generateRandomString(10);//生成随机初始明文
string key = generateRandomString(4);//随机生成初始key
string miwen = encryptionAPI(plainText, key);//得到初始加密密文
cout << miwen << endl;
int count = 1 << 10;
cout << count << endl;
string mingwen;
clock_t start = clock();//记录算法开始时间点
for (int i = 0; i < count; i++) {//检查对于随机明文、key对能不能有不同的key可以生成相同的密文
    vector<int> forceKey = forceInt2Binary(i);
    mingwen = forceEncryptionAPI(plainText, forceKey);
    if (mingwen == miwen) {//如果得到了相同的密文输出时间
        cout << "Success" << endl;
        clock_t end = clock();
        printf("算法运行了%d ms\n", end - start);
        start = end;
    }
}
```



循环结束只输出了一次成功，那么说明对于随机的明文密钥对不存在另一个 key 可以得到相同的密文。