1. **基本测试：**

基本GUI：密钥可以随机生成。



输入输出测试：



加密



解密

测试结果如上图所示，明文和密文一一对应，测试通过

1. **交叉测试：**



测试结果与其他小组同学相同，测试通过

1. **扩展功能：**

密钥沿用前例。同时需要注意，此处对于ASCII加密之后得到的密文采用UTF-8编码输出，原因是对于ASCII表，Java只能够展示出其中95个数据，剩下的数据乱码之后且不可复原，即不能够解密获得密文。因此采用其他编码进行输出。

**同密钥情况下加解密一段字符：**





如图所示，成功实现了对字符串的加密和解密，测试通过

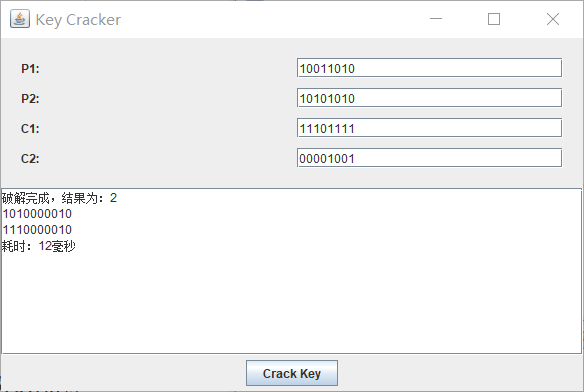
1. **给定明文密文对为：**

**10011010-11101111**

**10101010-00001001**

**加密密钥为1010000010**

**代入进程cheat.java求解：**



1. **推理为是，理由如下：**

**对于密钥生成两个子密钥，经过P10和P8只取决于除第二位之外的九位，因此对于上述密钥扩展置换，任意密钥均存在另一个密钥只需要满足第二位不同即可达成相同的加密解密效果。因此确实会出现选择不同的密钥得到相同密文的情况.**

**如下：将传入的10bit密钥加密后得到的**

**数字均代表位数。0表示第10位.**

**k1: 17948306**

**k2: 94836510**

**显然与第二位无关，。**