**测试报告**

1. **测试意义**
   1. **验证算法的正确性**

测试可以验证S-DES算法是否按照设计规格正确实现。这包括验证加密和解密过程的正确性，以及确保算法的输出符合预期。通过测试，可以评估S-DES算法的性能，包括加密和解密的速度，以及在不同环境和条件下算法的表现。

**1.2 发现潜在的缺陷和漏洞**

测试有助于发现S-DES算法实现中的潜在缺陷和漏洞，从而进行修复，提高算法的安全性和可靠性。通过全面的测试，可以提高S-DES算法实现的可信度，增加用户和开发者对算法的信任。

**1.3 确保算法的兼容性**

测试还可以确保S-DES算法在不同的平台和环境中能够正确执行，包括不同的操作系统、硬件配置和编程语言实现。测试结果可以为算法的进一步改进和优化提供依据，帮助开发者识别算法的瓶颈和改进点。

通过这些测试，可以确保S-DES算法实现的质量和可靠性，为算法的广泛应用提供基础。

1. **测试目的**

**2.1 功能测试**

验证S-DES算法实现的所有功能是否按照设计文档和规格说明书正确执行。这包括加密和解密功能、密钥生成和处理等。

**2.2 性能测试**

评估S-DES算法在不同条件下的性能表现，包括处理速度、资源消耗和响应时间等。

**2.3 兼容性测试**

确保S-DES算法在不同的平台和环境中能够正确执行，包括不同的操作系统、硬件配置和编程语言实现。

**2.4 回归测试**

在算法实现发生变更后，进行回归测试以确保新的变更没有引入新的缺陷，并且原有功能仍然正常工作。

**2.5 边界条件测试**

测试S-DES算法在边界条件下的表现，包括处理非常短或非常长的输入数据、异常输入等情况。

1. **测试步骤**
   1. **基本测试**
      1. 用户交互界面

实现了基本的GUI界面，可以在结果处展示“加密”和“解密”的结果。

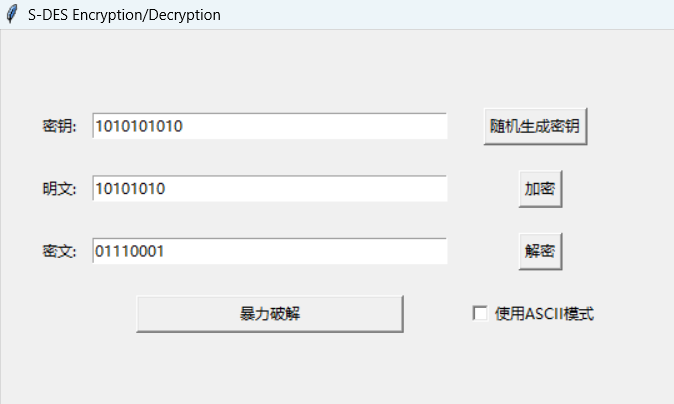
* + 1. 随机生成密钥

可以随机生成十位二进制密钥，也可以手动输入。

* + 1. 加密解密测试

本系统提供纯二进制加密解密功能及ASCⅡ编码字符串加密解密功能，在用户交互界面左上角可以选择，不勾选表示使用纯二进制数字进行加密解密。（此处选择不勾选）

①加密

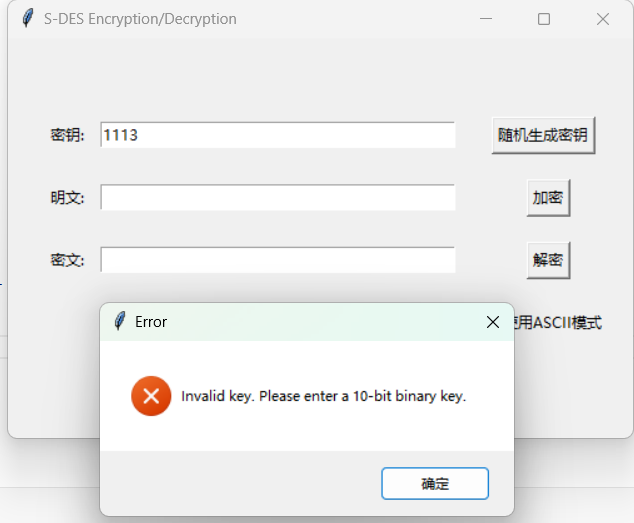


②解密



* + 1. 异常处理测试

当输入密钥或原文不符合规范时，会提供说明，需要重新填写。



* + 1. 测试总结

在本关卡中，我们小组成功完成了任务要求，主要做到了以下几点：

①通过实现GUI界面，我们学习了如何使用图形界面库（如Tkinter）来创建用户交互界面，以及如何处理用户输入和展示结果。

②通过实现密钥生成功能，我们学习了随机数生成和密钥处理的基本原理，以及如何在程序中实现这些功能。

③通过实现加密解密功能，我们学习了加密解密的基本原理，以及如何在程序中实现这些功能。我们还学习了如何处理不同类型的输入数据，包括纯二进制数字和ASCII编码字符串（在以下的关卡中）。

④通过实现异常处理功能，我们学习了如何在程序中处理异常情况，以及如何给出相应的提示信息，确保程序的健壮性和用户体验。

* 1. **交叉测试**

项目要求我们实现S-DES算法，并确保不同组同学编写的程序在使用相同的密钥时，对同一明文进行加密得到的密文相同，以及对同一密文进行解密得到的明文相同。这要求我们严格遵循S-DES算法的标准流程和转换单元（P-Box、S-Box等），以确保算法的正确性和一致性。

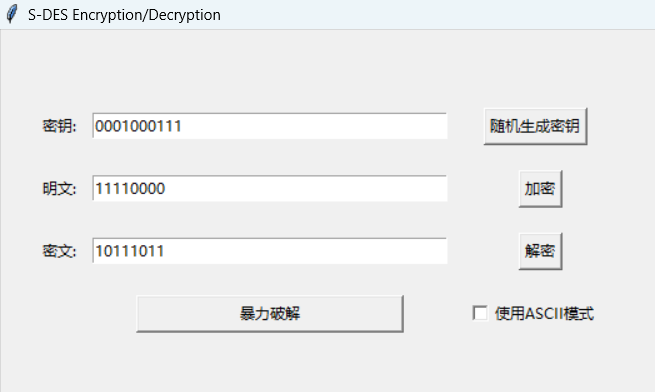
* + 1. A方主动加密，由B方进行协作测试；或B方主动加密，由A方进行协作测试

①二进制：

A方解密结果：密钥：0001000111

密文：10111011

求得明文：11110000



B方加密结果：密钥：0001000111

明文：11110000

求得密文：10111011



②ASCII编码字符串：

A方解密结果：密钥：1100110100

密文：ùÂg

求得明文：CQU



B方加密结果：密钥：1100110100

明文：CQU

求得密文：ùÂg



* + 1. 总结

在本关卡中，我们小组成功完成了任务要求，主要做到了以下几点：

①算法一致性：我们学习了如何确保不同实现的S-DES算法在加密和解密过程中保持一致性，即使在不同的系统或平台上运行；同时我们学习了如何在团队中进行协作测试，确保不同成员编写的程序能够正确地相互操作。

②加密解密流程及错误排查：我们深入理解了S-DES算法的加密和解密流程，包括P-Box和S-Box的转换原理和实现方法。在测试过程中，我们学习如何排查和解决不同实现之间的差异，确保算法的一致性和正确性。

* 1. **拓展功能**

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(可能是乱码)。本项目提供纯二进制加密解密功能及ASCⅡ编码字符串加密解密功能，在用户交互界面左上角可以选择，在本关卡将展示使用ASCⅡ编码字符串进行加密解密的功能。

* + 1. 随机生成密钥
    2. 加密解密测试

①加密：



②解密：



* + 1. 总结

在本关卡中，我们小组成功完成了任务要求，主要做到了以下几点：

①安全性考虑：我们测试了如何生成随机密钥，并将其应用于ASCII编码字符串的加密解密中，了解密钥在加密过程中的重要性，以及如何确保密钥的安全性。

②数据处理：我们学习了如何将ASCII编码字符串转换为二进制数据，并使用S-DES算法进行加密解密。在测试过程中，我们学习了如何处理加密解密过程中的异常情况，确保程序的健壮性。

③通过这些测试，我们不仅提高了对S-DES算法处理不同类型数据的能力，还加深了对加密解密过程的理解，为我们的编程技能和加密解密知识打下了坚实的基础。

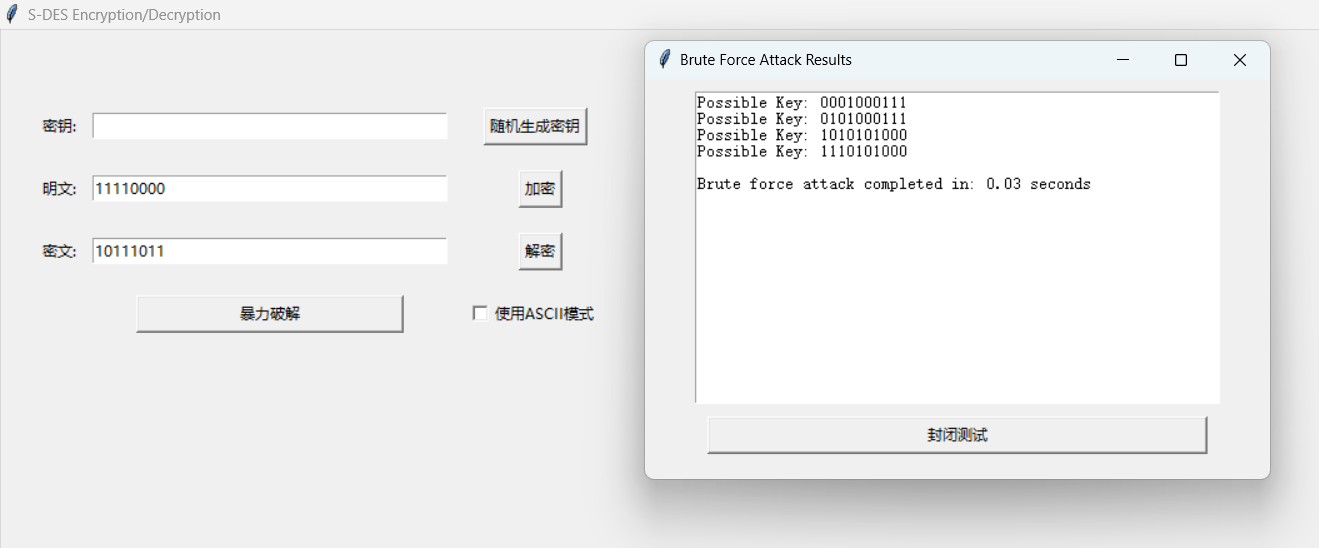
* 1. **暴力破解**

假设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。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。

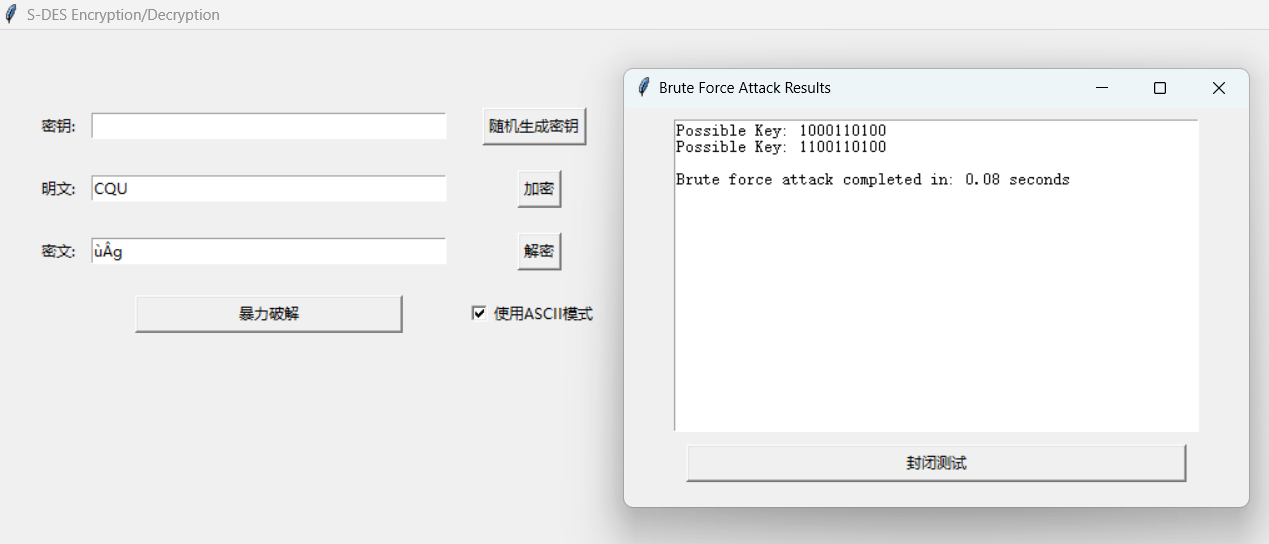
3.4.1 生成一组明密文对

3.4.2 破解测试

①二进制：



②ASCII编码字符串：



3.4.3 总结

在本关卡中，我们小组成功完成了任务要求，主要做到了以下几点：

**①暴力破解原理**：我们学习了暴力破解的基本原理，即尝试所有可能的密钥直到找到正确的密钥。我们实现了生成一组明密文对的功能，为暴力破解提供了测试数据。

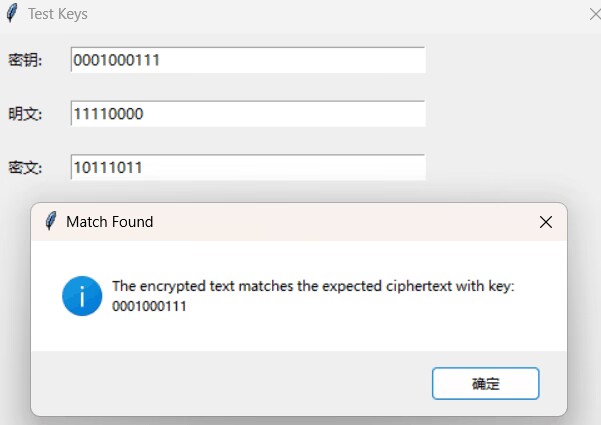
**②多线程编程**：我们学习了如何使用多线程技术来提升破解效率，包括线程的创建、同步和通信。

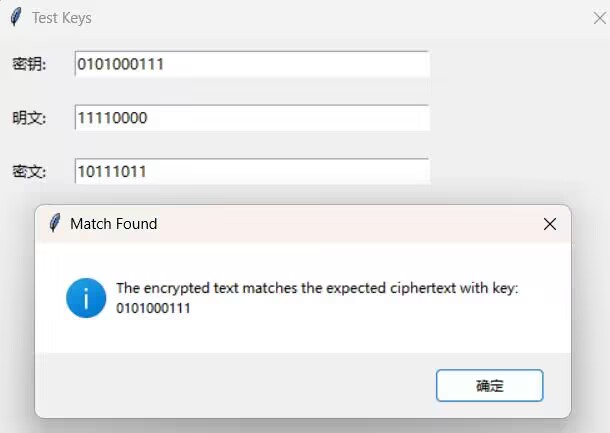
**③性能优化**：我们了解了如何通过优化算法和使用多线程来减少破解所需的时间。我们设定了时间戳，并使用视频或动图展示了在多长时间内完成了暴力破解，以证明破解的效率。

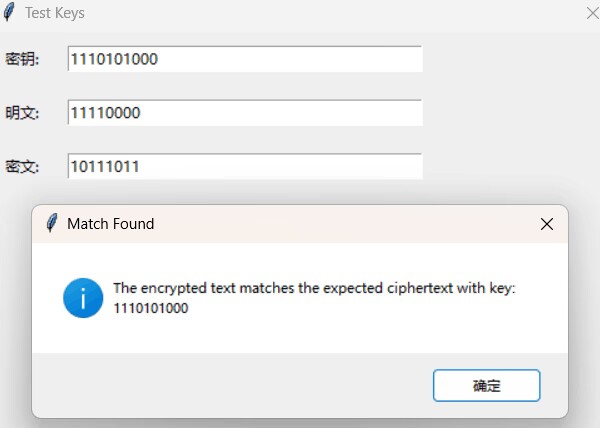
* 1. **封闭测试**

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，是否会出现选择不同的密钥K加密得到相同密文C的情况？

* + 1. **对于一个随机选择的明密文对，是否存在不止一个密钥Key？**







我们由此可以知道，通常情况下，对于一个给定的明文-密文对，只存在一个密钥Key 能够得到相应的密文C。而加密算法的目标是确保密文C与特定的密钥Key 相关联，以保护数据的安全性。因此，在现代加密算法中，密钥通常被设计为唯一的，以确保加密和解密的唯一性和可逆性。

**3.5.2 对应明文空间的任意给定明文分组P，是否会出现选择不同的密钥K加密得到相同密文C的情况？**

在理论上，对于不同的密钥K和相同的明文分组P，应该得到不同的密文C。这是因为加密算法的设计目的是确保不同密钥产生不同的密文，以增加破解的难度。然而，在一些特殊情况下，可能会出现不同密钥K产生相同密文C的情况，这通常是由于加密算法的缺陷或弱点所致。在现代密码学中，这种情况被视为严重的安全漏洞，因此加密算法的设计和评估都致力于防止这种情况的发生。

3.5.3 总结

在本关卡中，我们小组成功完成了任务要求，主要做到了以下几点：

①密钥设计原则：我们学习了现代加密算法中密钥通常被设计为唯一的，以确保加密和解密的唯一性和可逆性。我们通过测试验证了对于一个给定的明文-密文对，通常只存在一个密钥Key能够得到相应的密文C；同时在理论上，对于不同的密钥K和相同的明文分组P，应该得到不同的密文C。

②加密算法的安全性评估：我们评估了在特殊情况下，不同密钥K可能产生相同密文C的情况，并了解了这种情况被视为严重的安全漏洞。通过这些学习和实现，我们不仅加深了对S-DES算法密钥唯一性和加密算法目标的理解，还学习了现代加密算法的设计原则和安全性评估，为我们的编程技能和加密解密知识打下了坚实的基础。

1. **测试结果**

我们完成了基本的测试流程，以验证S-DES算法的实现是否正确，并确保其能够达到预定的目标，测试包括基本功能测试、性能测试、安全性测试等，以确保算法的正确性、效率和安全性。通过以上测试，我们证明了S-DES算法的实现是正确的，并且达到了预定的目标。算法在功能、性能和安全性方面都表现良好，可以用于实际的加密解密任务中。我们的测试结果表明，S-DES算法是一个可靠和高效的加密工具，可以用于保护数据的安全性。