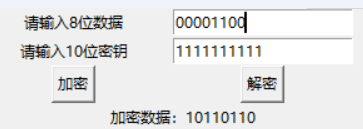
# 测试结果：

## 第一关：

加密操作：

明文:00001100 密钥:1111111111

加密结果:10110110



解密操作：

密文:00001111 密钥:1000100010

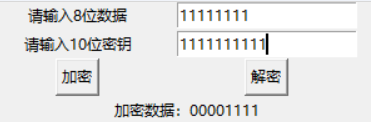
解密结果:11100111

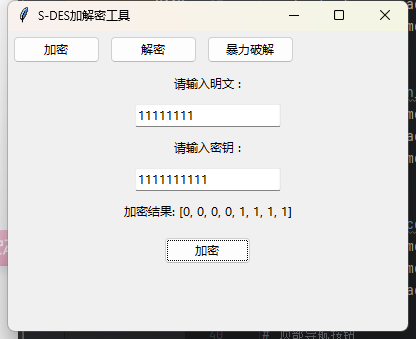


## 第二关：

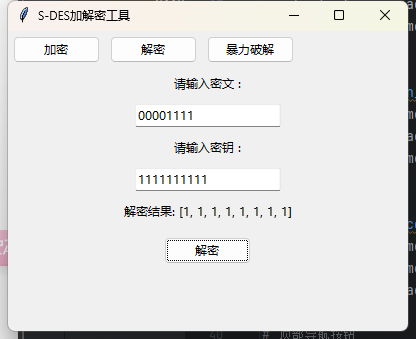
与其他小组交叉测试结果：

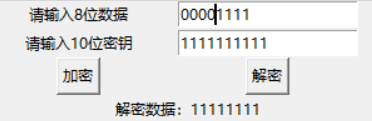
加密：





解密：

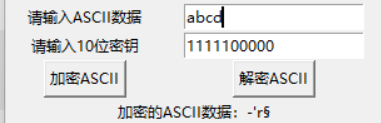




## 第三关：

对ASCII字符串进行加密

密钥：1111100000 明文：abcd



## 第四关：

暴力破解：



**第五关：**

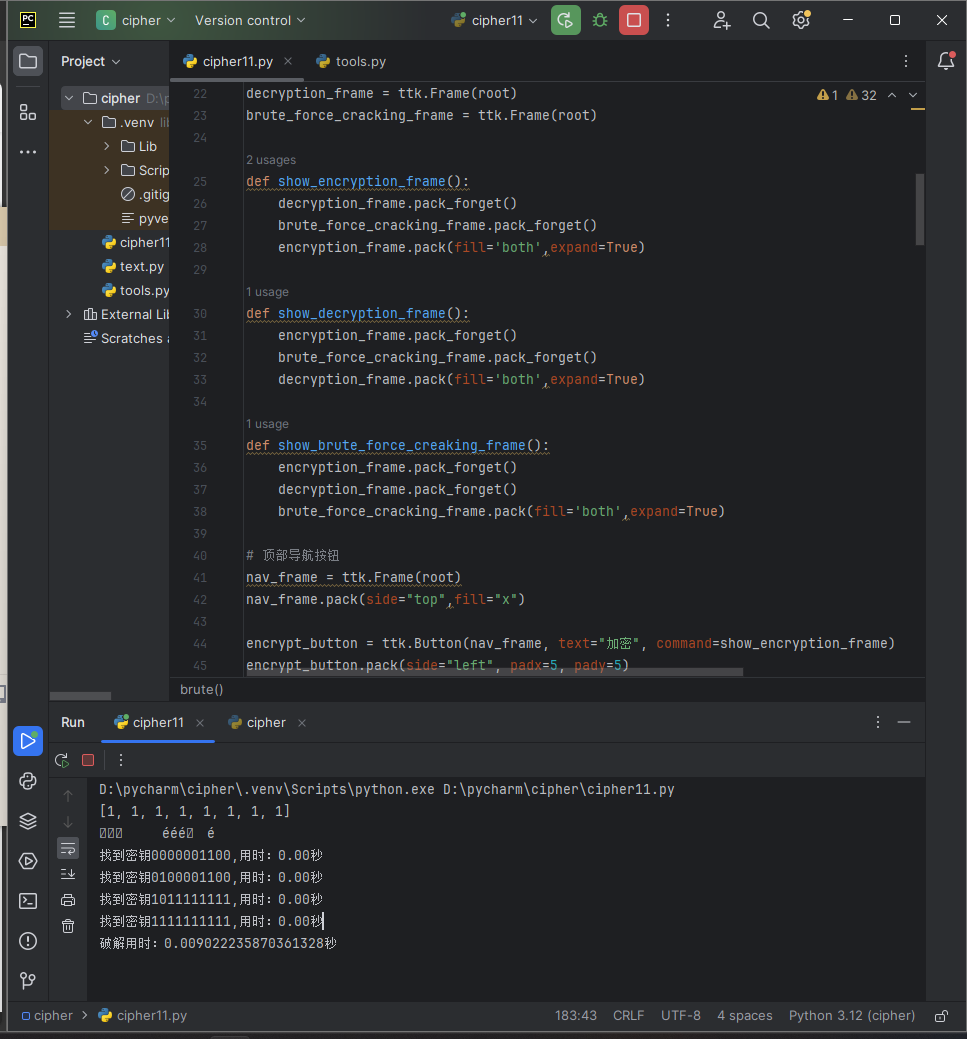
暴力破解函数：

def brute\_force():  
 cipher\_text = brute\_force\_cipher\_entry.get()  
 known\_plaintext = brute\_force\_plain\_entry.get()  
  
 if not (len(cipher\_text) == 8 and len(known\_plaintext) == 8 and  
 all(bit in '01' for bit in cipher\_text + known\_plaintext)):  
 messagebox.showerror("输入错误", "请输入8位二进制密文和8位二进制明文。")  
 return  
  
 found\_keys = [] # 用于存储可能的密钥  
 progress = 0  
  
 for i in range(1024): # 生成所有可能的10位密钥  
 key = f'{i:010b}' # 将i转换为10位二进制格式  
 decrypted\_text = encrypt\_decrypt(cipher\_text, key, encrypt=False)  
  
 # 显示当前正在尝试的密钥  
 brute\_force\_output.config(text=f"尝试密钥: {key}")  
 root.update()  
  
 if decrypted\_text == known\_plaintext: # 如果解密结果与已知明文一致  
 found\_keys.append(key)  
  
 progress += 1  
 progress\_label.config(text=f"进度: {progress}/1024")  
  
 # 显示结果  
 if found\_keys:  
 brute\_force\_output.config(text=f"可能的密钥：{', '.join(found\_keys)}")  
 else:  
 brute\_force\_output.config(text="没有找到合适的密钥。")

我们的破解函数能找到一对明密文之间的所有可能密钥，

例如：

明文：11111111 密文：00001111



发现该明密文之间有四把不同的密钥，破解花费时间为0.009秒