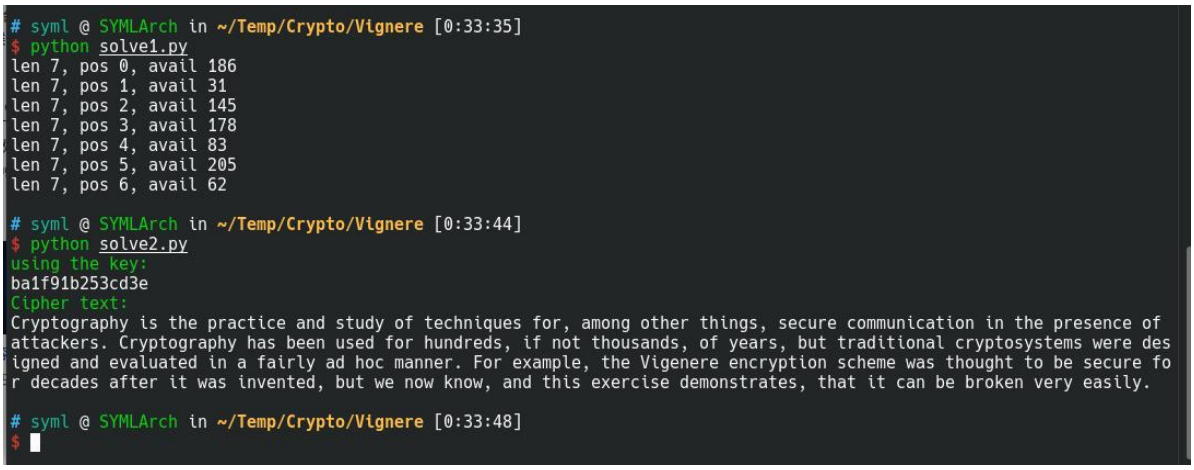
```
using the key:
ba1f91b253cd3e
Cipher text:
Cryptography is the practice and study of techniques for, among other things, secure communication in the presence of attackers. Cryptography has been used for hundreds, if not thousands, of years, but traditional cryptosystems were designed and evaluated in a fairly ad hoc manner. For example, the Vigenere encryption scheme was thought to be secure for decades after it was invented, but we now know, and this exercise demonstrates, that it can be broken very easily.
```




```python
# 密钥推算，本来预期有多组可能的结果，但实际只有一组

from binascii import a2b_hex
from Crypto.Util.number import long_to_bytes
import re

cipher = a2b_hex(b"F96DE8C227A259C87EE1DA2AED57C93FE5DA36ED4EC87EF2C63AAE5B9A7EFFD673BE4ACF7BE8923CAB1ECE7AF2DA3DA44FCF7AE29235A24C963FF0DF3CA3599A70E5DA36BF1ECE77F8DC34BE129A6CF4D126BF5B9A7CFEDF3EB850D37CF0C63AA2509A76FF9227A55B9A6FE3D720A850D97AB1DD35ED5FCE6BF0D138A84CC931B1F121B44ECE70F6C032BD56C33FF9D320ED5CDF7AFF9226BE5BDE3FF7DD21ED56CF71F5C036A94D963FF8D473A351CE3FE5DA3CB84DDB71F5C17FED51DC3FE8D732BF4D963FF3C727ED4AC87EF5DB27A451D47EFD9230BF47CA6BFEC12ABE4ADF72E29224A84CDF3FF5D720A459D47AF59232A35A9A7AE7D33FB85FCE7AF5923AA31EDB3FF7D33ABF52C33FF0D673A551D93FFCD33DA35BC831B1F43CBF1EDF67F0DF23A15B963FE5DA36ED68D378F4DC36BF5B9A7AFFD121B44ECE76FEDC73BE5DD27AFCD773BA5FC93FE5DA3CB859D26BB1C63CED5CDF3FE2D730B84CDF3FF7DD21ED5ADF7CF0D636BE1EDB79E5D721ED57CE3FE6D320ED57D469F4DC27A85A963FF3C727ED49DF3FFFDD24ED55D470E69E73AC50DE3FE5DA3ABE1EDF67F4C030A44DDF3FF5D73EA250C96BE3D327A84D963FE5DA32B91ED36BB1D132A31ED87AB1D021A255DF71B1C436BF479A7AF0C13AA14794")

available_text = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz,.;'?!: "



def decrypt_by_bit(cipher: bytes, pos: int, key: int, key_len: int):
    for i in range(len(cipher) // key_len):
        if chr(cipher[i * key_len + pos] ^ key) not in available_text:
```

```
            return False
    return True


for key_len in range(2, 13):
    for key_byte in range(key_len):
        for key in range(2 ** 8):
            if decrypt_by_bit(cipher, key_byte, key, key_len):
                print(f"len {key_len}, pos {key_byte}, avail {key}")

```

```
# 解密

from binascii import a2b_hex
from Crypto.Util.number import long_to_bytes
import re

cipher =
a2b_hex(b"F96DE8C227A259C87EE1DA2AED57C93FE5DA36ED4EC87EF2C63AAE5B9A7EFFD673B
E4ACF7BE8923CAB1ECE7AF2DA3DA44FCF7AE29235A24C963FF0DF3CA3599A70E5DA36BF1ECE77
F8DC34BE129A6CF4D126BF5B9A7CFEDF3EB850D37CF0C63AA2509A76FF9227A55B9A6FE3D720A
850D97AB1DD35ED5FCE6BF0D138A84CC931B1F121B44ECE70F6C032BD56C33FF9D320ED5CDF7A
FF9226BE5BDE3FF7DD21ED56CF71F5C036A94D963FF8D473A351CE3FE5DA3CB84DDB71F5C17FE
D51DC3FE8D732BF4D963FF3C727ED4AC87EF5DB27A451D47EFD9230BF47CA6BFEC12ABE4ADF72
E29224A84CDF3FF5D720A459D47AF59232A35A9A7AE7D33FB85FCE7AF5923AA31EDB3FF7D33AB
F52C33FF0D673A551D93FFCD33DA35BC831B1F43CBF1EDF67F0DF23A15B963FE5DA36ED68D378
F4DC36BF5B9A7AFFD121B44ECE76FEDC73BE5DD27AFCD773BA5FC93FE5DA3CB859D26BB1C63CE
D5CDF3FE2D730B84CDF3FF7DD21ED5ADF7CF0D636BE1EDB79E5D721ED57CE3FE6D320ED57D469
F4DC27A85A963FF3C727ED49DF3FFFDD24ED55D470E69E73AC50DE3FE5DA3ABE1EDF67F4C030A
44DDF3FF5D73EA250C96BE3D327A84D963FE5DA32B91ED36BB1D132A31ED87AB1D021A255DF71
B1C436BF479A7AF0C13AA14794")


key = [186, 31, 145, 178, 83, 205, 62]
print("\033[32musing the key:\033[0m")
for i in key:
    k = hex(i)[2:]
    if len(k)!=2:
        k = '0'+k
    print(k, end="")

def decrypt(cipher: bytes, key: list):
    plain_text = ""
    for i in range(len(cipher)):
        plain_text += chr(key[i % len(key)] ^ cipher[i])
    if plain_text.find("the") != -1:
        return True, plain_text
    return False, plain_text

state, ans = decrypt(cipher, key)
print(f"\n\033[32mCipher text:\033[0m\n{ans}")

```