Student name: Edvinas Dulskas
Student code: 19040186

source code ***shift_cipher.py***

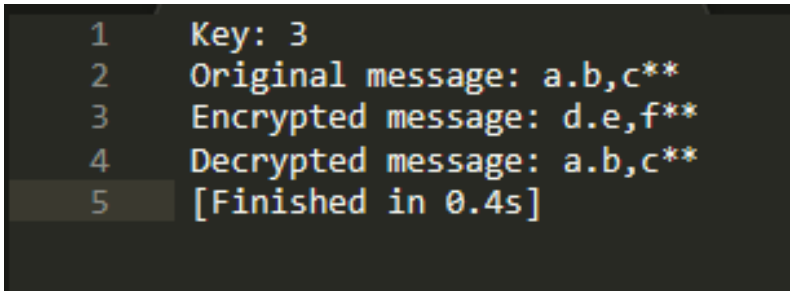
```
from string import ascii_lowercase as lower
#from string import ascii_uppercase as upper


# -------- Functions ------------
# function that encrypts the given message
def shift_encrypt(key, msg):

        temp = msg.lower()
        ctx_indexes = [] # empty list for mod26 indexes
        ctx = ''                                        # emty string for ciphertext

        for s in temp:
                if s in table:
                        idx = get_mod26(key, table[s], 'e')             # get mo26 values
                        ctx_indexes.append(idx)
        # add mod23 values to the list
                else:
                        ctx_indexes.append(s)

        # encrypt the message
        for i in ctx_indexes:
                if i in table.values():
                        ctx += list(table.keys())[list(table.values()).index(i)]                # get
key from dict by it's value and add it to the string
                else:
                        ctx += i

        return ctx

# function that decrypts the encrypted message
def shift_decrypt(key, ctx):
```

```
        temp = ctx
        pp_indexes = []                 # empty list for mod26 indexes
        pp = ''                                                 # emty string for decrypted message

        for s in temp:
                if s in table:
                        idx = get_mod26(key, table[s], 'd')             # get mo26 values
                        pp_indexes.append(idx)
        # add mod23 values to the list
                else:
                        pp_indexes.append(s)

        # decrypt the ciphertext
        for i in pp_indexes:
                if i in table.values():
                        pp += list(table.keys())[list(table.values()).index(i)]                 # get
key from dict by it's value and add it to the string
                else:
                        pp += i

        return pp

# funtion that creates a dictionary
def create_dictionary():

        temp = {}

        for i in lower:
                temp[i] = len(temp)

        return temp

# function that returns mod26 number
```

Student name: Edvinas Dulskas
Student code: 19040186

```
def get_mod26(key, number, method):

        temp = 0

        if method == 'e':
                temp = (number + key) % 26
        if method == 'd':
                temp = number - key
                if temp < 0:
                        temp += 26

        return temp

# --------- Main code ----------
table = create_dictionary()
key = 3
msg = "a.b,c**"

C = shift_encrypt(key, msg)
P = shift_decrypt(key, C)

print('Key: ' + str(key) + '\nOriginal message: ' + msg + '\nEncrypted message: ' + C + '\nDecrypted message: ' + P)
```

*Results of **program**:*



Explanation: the key is 3 as shown in the image. The thing that I did was to ignor the spaces and special characters such as „,./*-+\...“ during the encryption.