cryptography

Cryptography - 문승재

Go!

Correct!!

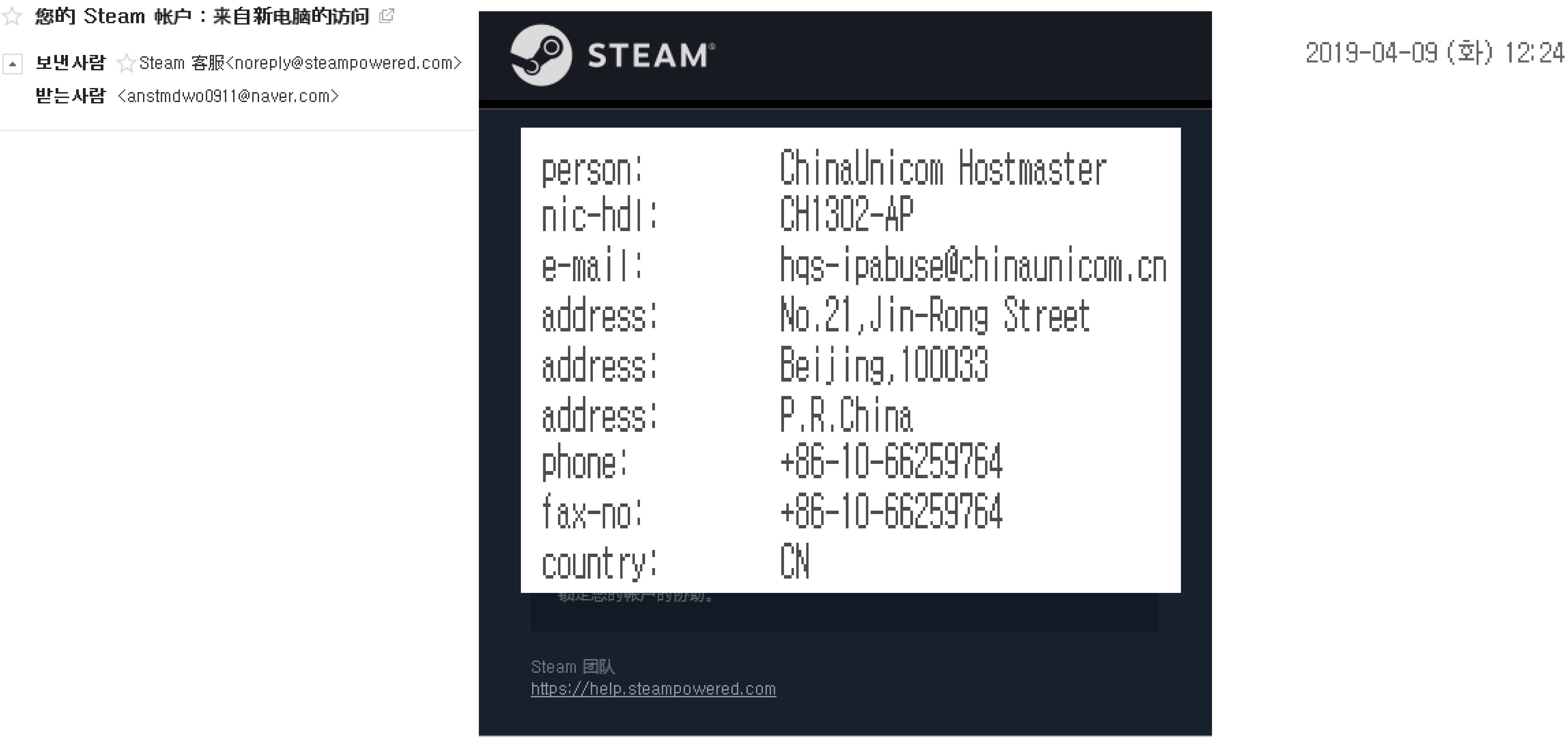
목 차

- 암호 공부를 시작하게 된 이유
- 앞으로의 발표 계획
- 암호의 종류
- CAESAR암호
- VIGENERE암호
- 코드 설명

암호를 시작 하게 된 계기

암호를 시작 하게 된 계기

Part 1
Part 2
Part 3
Part 4
Part 5
Part 6
Part 7



☆ 您的 Steam 帐户：来自新电脑的访问

보낸사람 ☆Steam 客服<noreply@steampowered.com>
받는사람 <anstmdwo0911@naver.com>

2019-04-09 (화) 12:24

```
person:        ChinaUnicom Hostmaster
nic-hdl:       CH1302-AP
e-mail:        hqs-ipabuse@chinaunicom.cn
address:       No.21,Jin-Rong Street
address:       Beijing,100033
address:       P.R.China
phone:         +86-10-66259764
fax-no:        +86-10-66259764
country:       CN
```

锁定您的帐户的协助。

Steam 团队
https://help.steampowered.com

암호를 시작 하게 된 계기

Part 1
Part 2
Part 3
Part 4
Part 5
Part 6
Part 7

앞으로의 발표 계획

Part 1
Part 2
Part 3
Part 4
Part 5
Part 6
Part 7

암호의 종류

코드 설명

DES암호

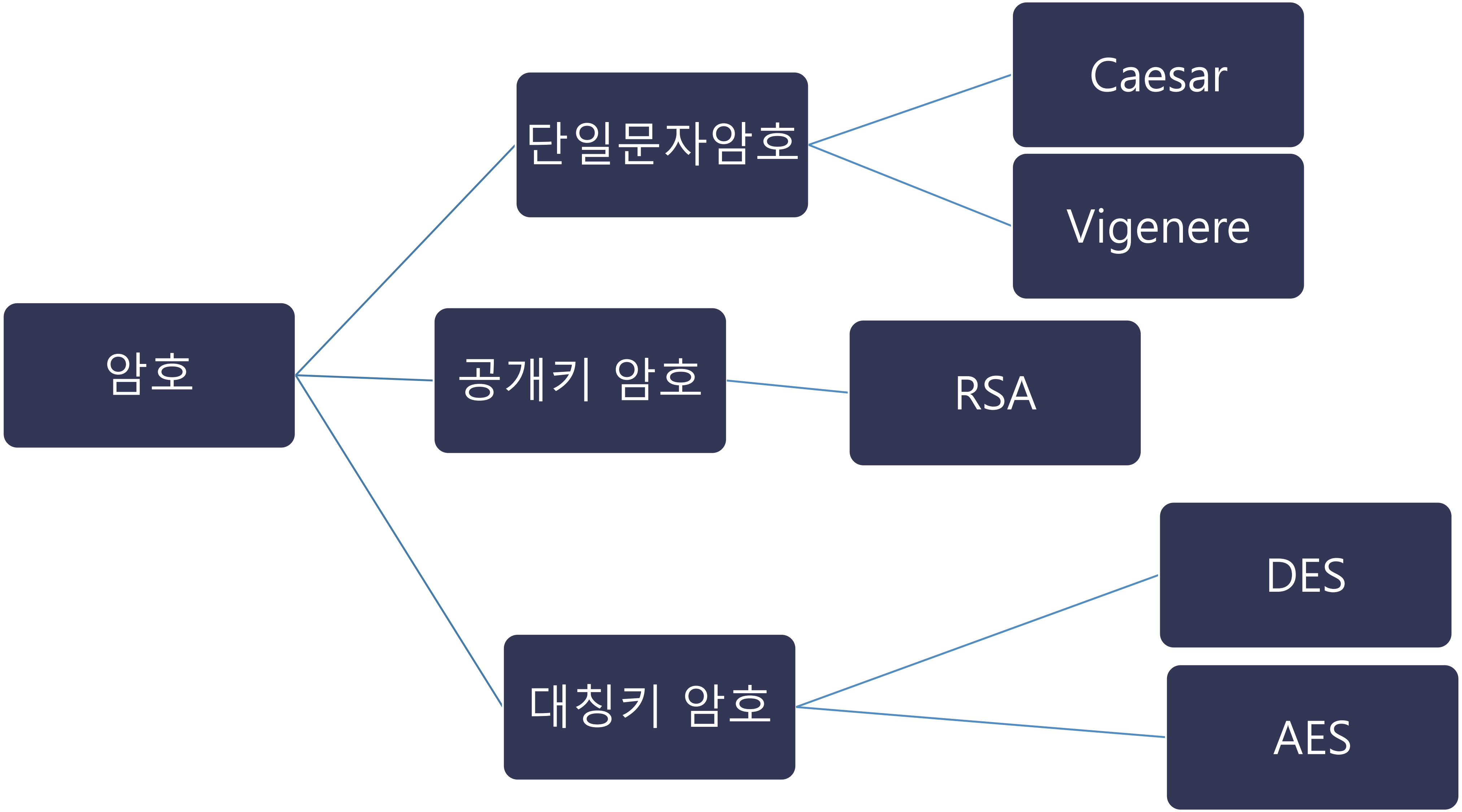
or

RSA암호

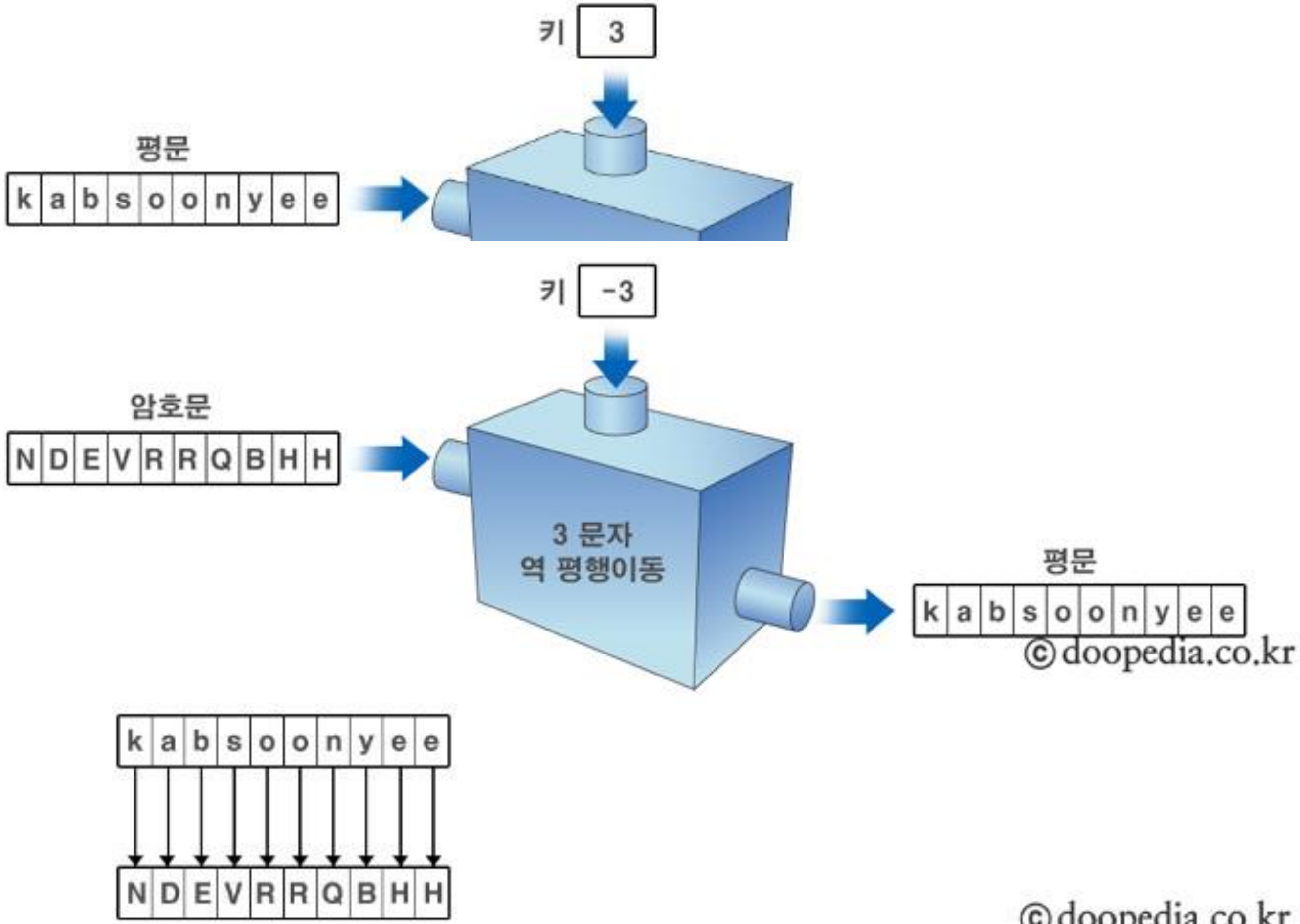
프로젝트

발표

암호의 종류



Caecar cipher



© doopedia.co.kr

© doopedia.co.kr

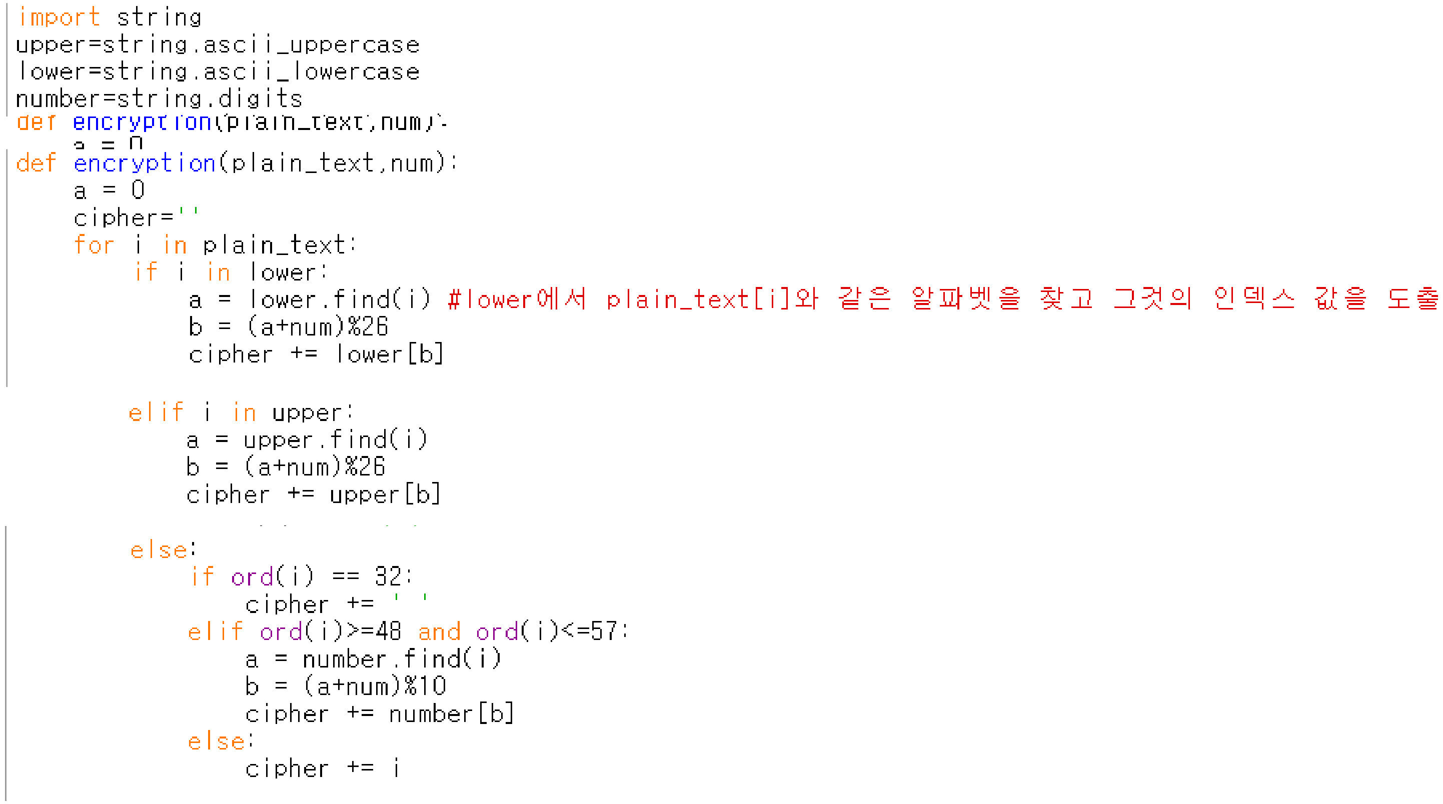
Vigenere cipher

Part 1
Part 2
Part 3
Part 4
Part 5
Part 6
Part 7

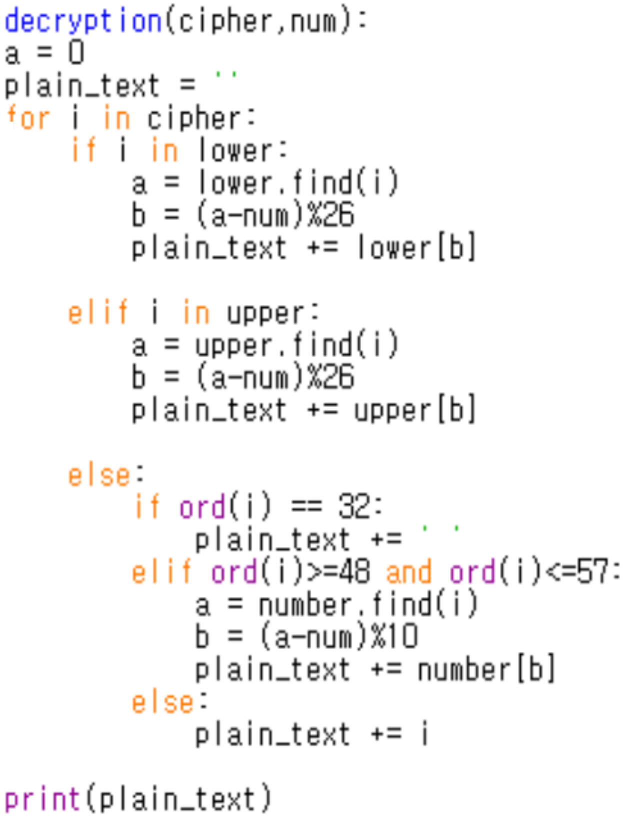
원문	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
1	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
2	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
3	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
4	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
5	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
6	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
7	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
8	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
9	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
10	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
11	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
18	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
19	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
20	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
21	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
22	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
23	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
24	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
25	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
26	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

키워드	S	K	Y	S	K	Y	S	K	Y	S	K	Y	S	K	Y	S	K	Y	S	K	Y	S	K
원문	d	i	v	e	r	t	t	r	o	o	p	s	t	o	e	a	s	t	r	i	d	g	e
암호문	V	S	T	W	B	R	L	B	M	G	Z	Q	L	Y	C	S	C	R	J	S	B	Y	O

Caesar코드 설명 - 문장암호화

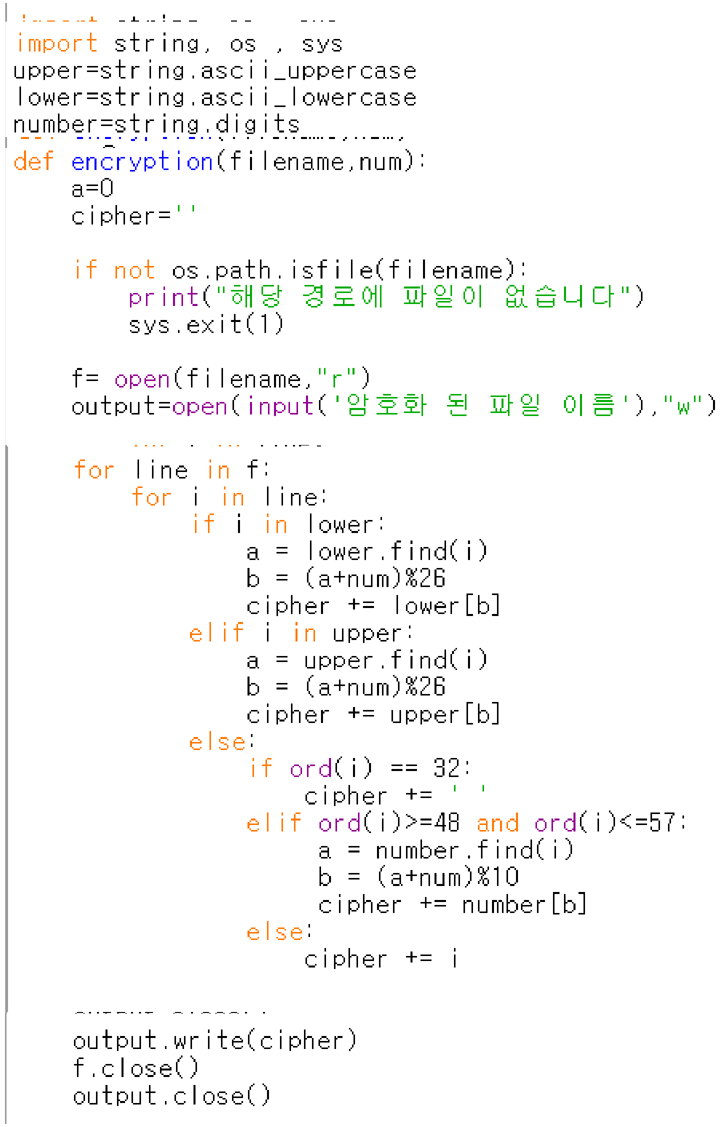


```
import string
upper=string.ascii_uppercase
lower=string.ascii_lowercase
number=string.digits
def encryption(plain_text,num):
    a = 0
def encryption(plain_text,num):
    a = 0
    cipher=''
    for i in plain_text:
        if i in lower:
            a = lower.find(i) #lower에서 plain_text[i]와 같은 알파벳을 찾고 그것의 인덱스 값을 도출
            b = (a+num)%26
            cipher += lower[b]

        elif i in upper:
            a = upper.find(i)
            b = (a+num)%26
            cipher += upper[b]

        else:
            if ord(i) == 32:
                cipher += ' '
            elif ord(i)>=48 and ord(i)<=57:
                a = number.find(i)
                b = (a+num)%10
                cipher += number[b]
            else:
                cipher += i
```

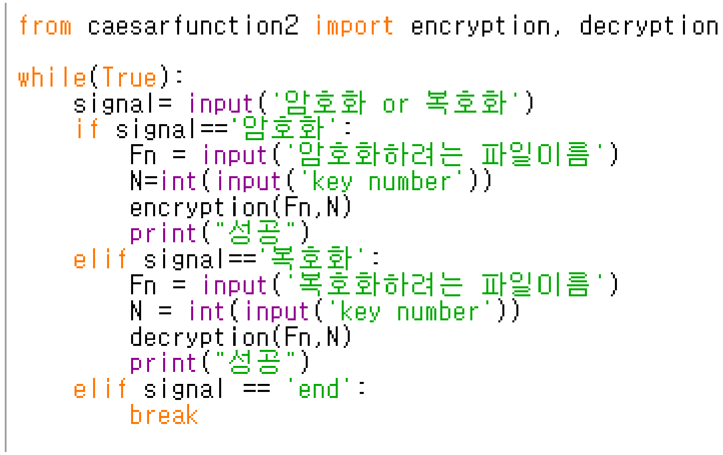


```
decryption(cipher,num):
a = 0
plain_text = ''
for i in cipher:
    if i in lower:
        a = lower.find(i)
        b = (a-num)%26
        plain_text += lower[b]

    elif i in upper:
        a = upper.find(i)
        b = (a-num)%26
        plain_text += upper[b]

    else:
        if ord(i) == 32:
            plain_text += ' '
        elif ord(i)>=48 and ord(i)<=57:
            a = number.find(i)
            b = (a-num)%10
            plain_text += number[b]
        else:
            plain_text += i

print(plain_text)
```

Caesar코드 설명 - 파일암호화



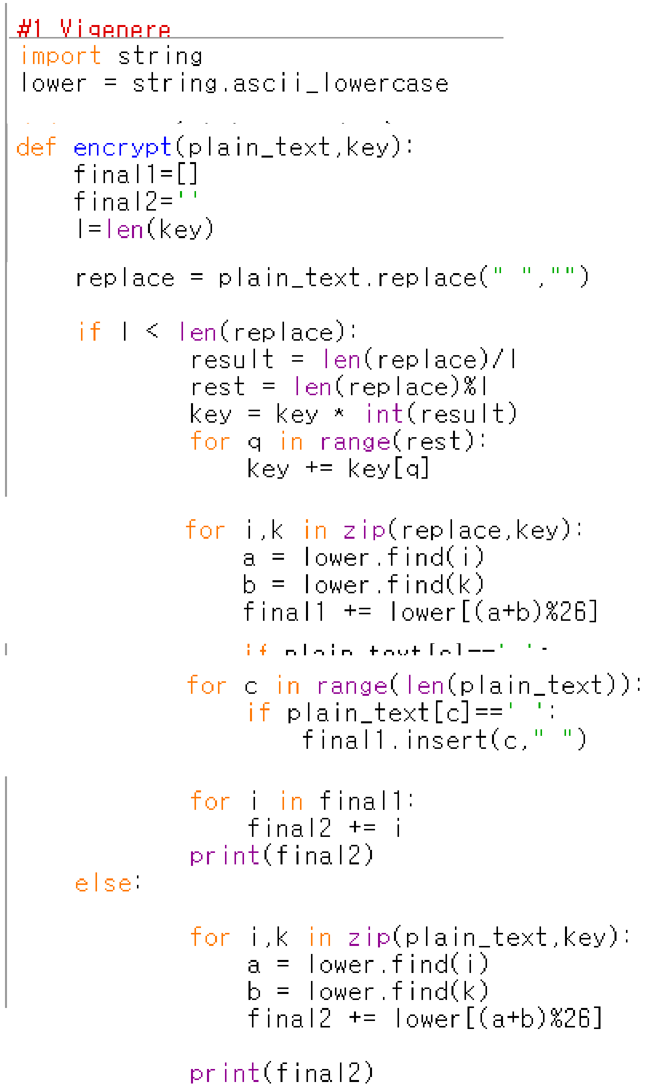
```
import string, os , sys
upper=string.ascii_uppercase
lower=string.ascii_lowercase
number=string.digits
def encryption(filename,num):
    a=0
    cipher=''

    if not os.path.isfile(filename):
        print("해당 경로에 파일이 없습니다")
        sys.exit(1)

    f= open(filename,"r")
    output=open(input('암호화 된 파일 이름'),"w")

    for line in f:
        for i in line:
            if i in lower:
                a = lower.find(i)
                b = (a+num)%26
                cipher += lower[b]
            elif i in upper:
                a = upper.find(i)
                b = (a+num)%26
                cipher += upper[b]
            else:
                if ord(i) == 32:
                    cipher += ' '
                elif ord(i)>=48 and ord(i)<=57:
                     a = number.find(i)
                     b = (a+num)%10
                     cipher += number[b]
                else:
                    cipher += i

    output.write(cipher)
    f.close()
    output.close()
```



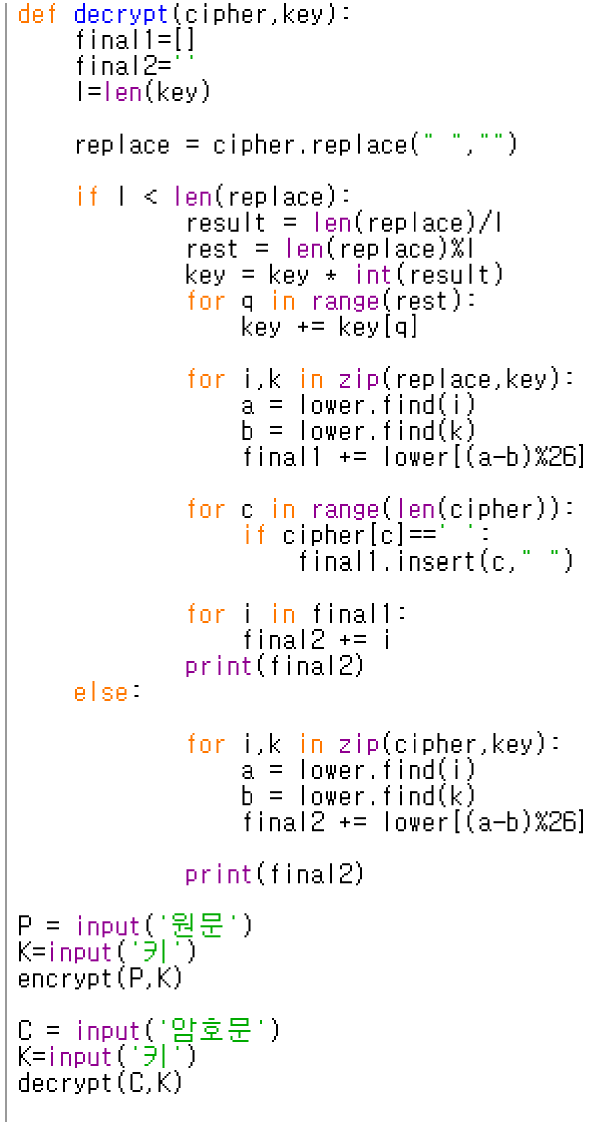
```
from caesarfunction2 import encryption, decryption

while(True):
    signal= input('암호화 or 복호화')
    if signal=='암호화':
        Fn = input('암호화하려는 파일이름')
        N=int(input('key number'))
        encryption(Fn,N)
        print("성공")
    elif signal=='복호화':
        Fn = input('복호화하려는 파일이름')
        N = int(input('key number'))
        decryption(Fn,N)
        print("성공")
    elif signal == 'end':
        break
```

Vigenere코드 설명

Part 1
Part 2
Part 3
Part 4
Part 5
Part 6
Part 7



```
#1 Vigenere
import string
lower = string.ascii_lowercase

def encrypt(plain_text,key):
    final1=[]
    final2=''
    l=len(key)

    replace = plain_text.replace(" ","")

    if l < len(replace):
            result = len(replace)/l
            rest = len(replace)%l
            key = key * int(result)
            for q in range(rest):
                key += key[q]

            for i,k in zip(replace,key):
                a = lower.find(i)
                b = lower.find(k)
                final1 += lower[(a+b)%26]

                if plain_text[c]==' ':
            for c in range(len(plain_text)):
                if plain_text[c]==' ':
                    final1.insert(c," ")

            for i in final1:
                final2 += i
            print(final2)
    else:

            for i,k in zip(plain_text,key):
                a = lower.find(i)
                b = lower.find(k)
                final2 += lower[(a+b)%26]

            print(final2)
```



```
def decrypt(cipher,key):
    final1=[]
    final2=''
    l=len(key)

    replace = cipher.replace(" ","")

    if l < len(replace):
            result = len(replace)/l
            rest = len(replace)%l
            key = key * int(result)
            for q in range(rest):
                key += key[q]

            for i,k in zip(replace,key):
                a = lower.find(i)
                b = lower.find(k)
                final1 += lower[(a-b)%26]

            for c in range(len(cipher)):
                if cipher[c]==' ':
                  final1.insert(c," ")

            for i in final1:
                final2 += i
            print(final2)
    else:

            for i,k in zip(cipher,key):
                a = lower.find(i)
                b = lower.find(k)
                final2 += lower[(a-b)%26]

            print(final2)

P = input('원문')
K=input('키')
encrypt(P,K)

C = input('암호문')
K=input('키')
decrypt(C,K)
```

Part 1
Part 2
Part 3
Part 4
Part 5
Part 6
Part 7

감사합니다!