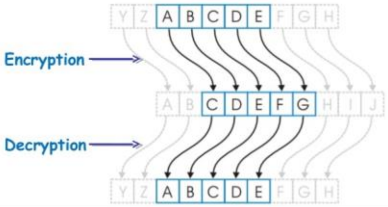
X	Y	Z	A	B	C	Unicode Character	a → z
88	89	90	65	66	67	Unicode Ordinal	97 → 122
⬇ minus ord("A") ⬇			⬆ plus ord("A") ⬆				⬇ minus ord('a') ⬇ ⬆ plus ord('a') ⬆
23	24	25	0	1	2	"Base 26" Ordinal	
						equal to	
23%26	24%26	25%26	0%26	1%26	2%26	Modulo	

How to use modulo to "Wrap" on base 26: or 26%26 or 27%26 or 28%26

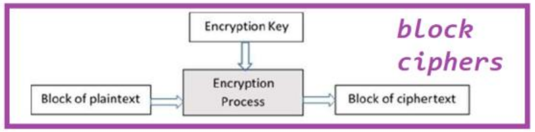
Symmetric cryptography (both *encrpytion* and *decryption*) uses the **same** key

Caeser shift:



Caesar shift, polyaphabetic substitution (e.g. Vignere and Gronsfeld), and one-time pad encryption algorithms you may have to write. Simple block cipher algorithms you may also have to write.

Hmmmmmm. Expect this on the exam you must.



block ciphers

plain text	W	I	Z
one-time pad key	A (0)	B (1)	C (2)
cipher text	W	J	B

hashing: one-way function resulting in a unique number (the resulting unique number is known as a hash digest or checksum)

Commerical Hash Algorithms: MD5, SHA-1, SHA-2

```
import hashlib
hashlib.md5(b'password').digest()
```

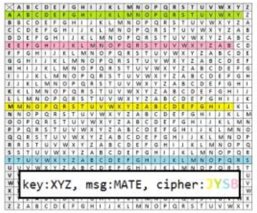
```
import hashlib
import random
alphabet = ["a","b","c","d","e","f","g"]
password ='password'.encode()
salt = "".join(random.sample(alphabet,5)).encode()
digest = hashlib.md5(salt+password).hexdigest()
```

salt: unique random value added to plain text *before hashing*, which prevents the same plain text values from generating the same hash digest.

Gronsfeld (numeric key):

plain text	A	B	C	D
key	1	2	3	
cipher text	B	D	F	E

Vignere:



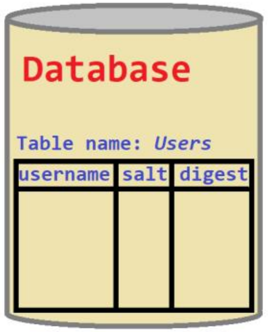
⊠	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	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
G	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
H	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
I	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
J	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
K	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
L	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
M	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	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	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
S	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
T	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
U	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
V																										U
W																										V
X																										W
Y																										X
Z	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

key:XYZ, msg:MATE, cipher:JYSB

Commercial Symmetric Encryption Algorithms: Data Encryption Standard (DES), Triple DES, Advanced Encryption Standard (AES), Blowfish and Twofish

can you recognise and describe?

Commercial Asymmetric Algorithms:
- RSA
- DSA



Asymmetric cryptography uses **different** keys. A *public key* is used to encrypt. A *private key* is used to decrypt. Both keys are different but 'linked' mathematically. This is often used in the transmission of data between a secured section of a website and an end user.