Cryptography

Section 2

Created by/ Rehab Mohamed & Hanaa Mansour

Substitution Technique in Cryptography

- Substitution technique is a classical encryption technique where the characters present in the original message are replaced by the other characters or numbers or by symbols.

Substitution Technique:

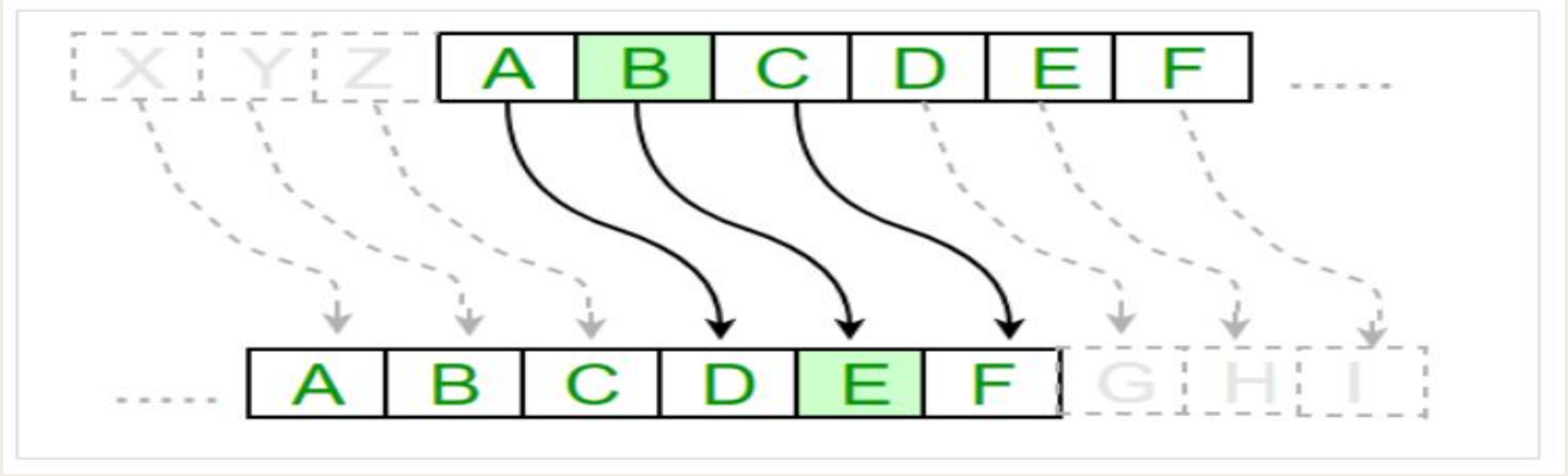
1. Caesar Cipher
2. Monoalphabetic Cipher
3. Playfair Cipher
4. Hill Cipher
5. Polyalphabetic Cipher

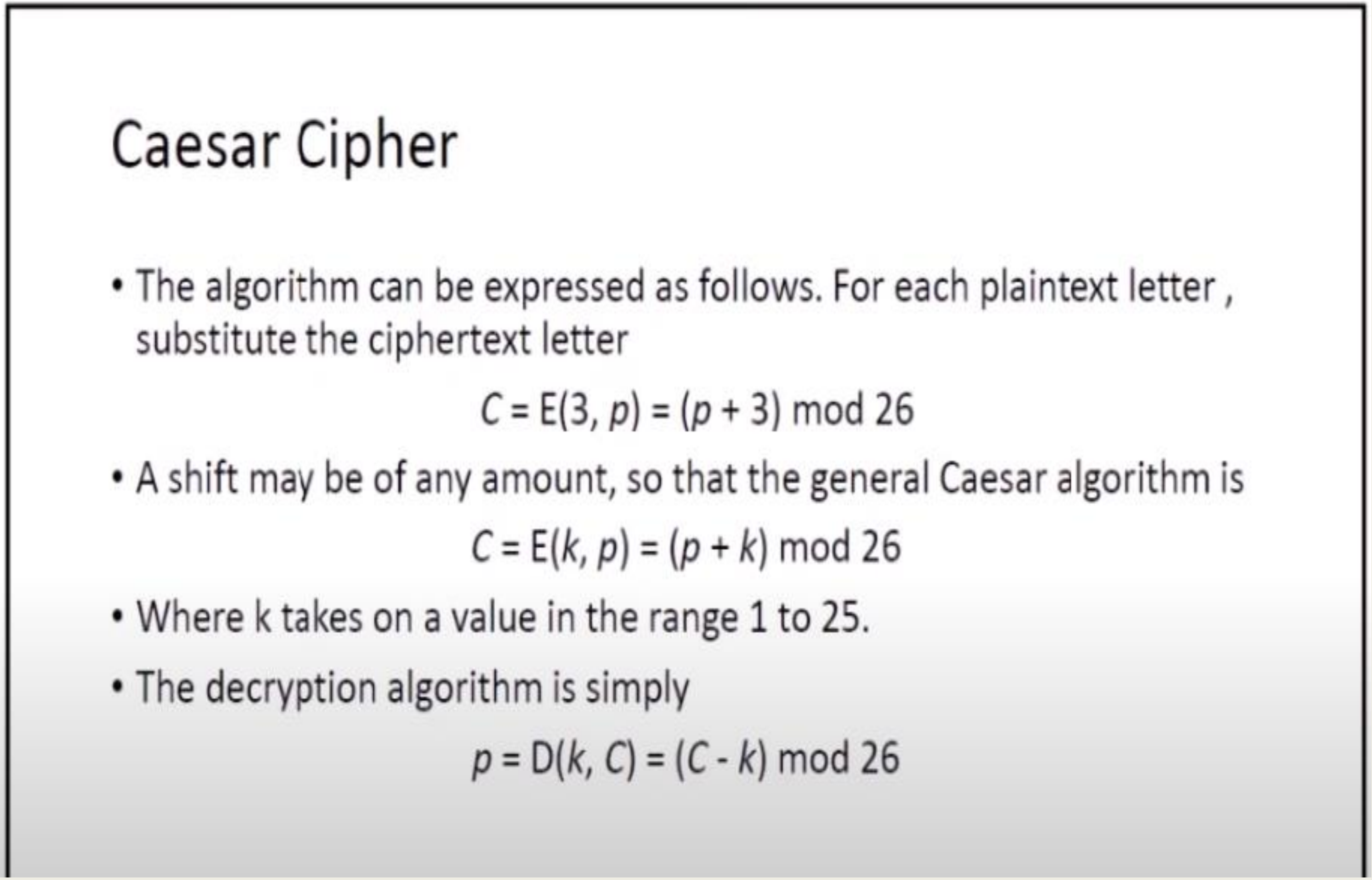
Caesar Cipher

Algorithm of Caesar Cipher

The algorithm of Caesar cipher holds the following features:

- Caesar Cipher Technique is the simple and easy method of encryption technique.
- It is simple type of substitution cipher.
- Each letter of plain text is replaced by a letter with some fixed number of positions down with alphabet.





Caesar Cipher

- The algorithm can be expressed as follows. For each plaintext letter , substitute the ciphertext letter

$$C = \mathrm{E}(3, p) = (p + 3) \bmod 26$$

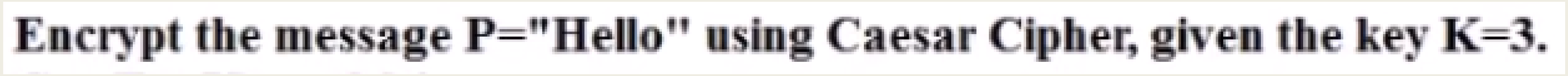
- A shift may be of any amount, so that the general Caesar algorithm is

$$C = \mathrm{E}(k, p) = (p + k) \bmod 26$$

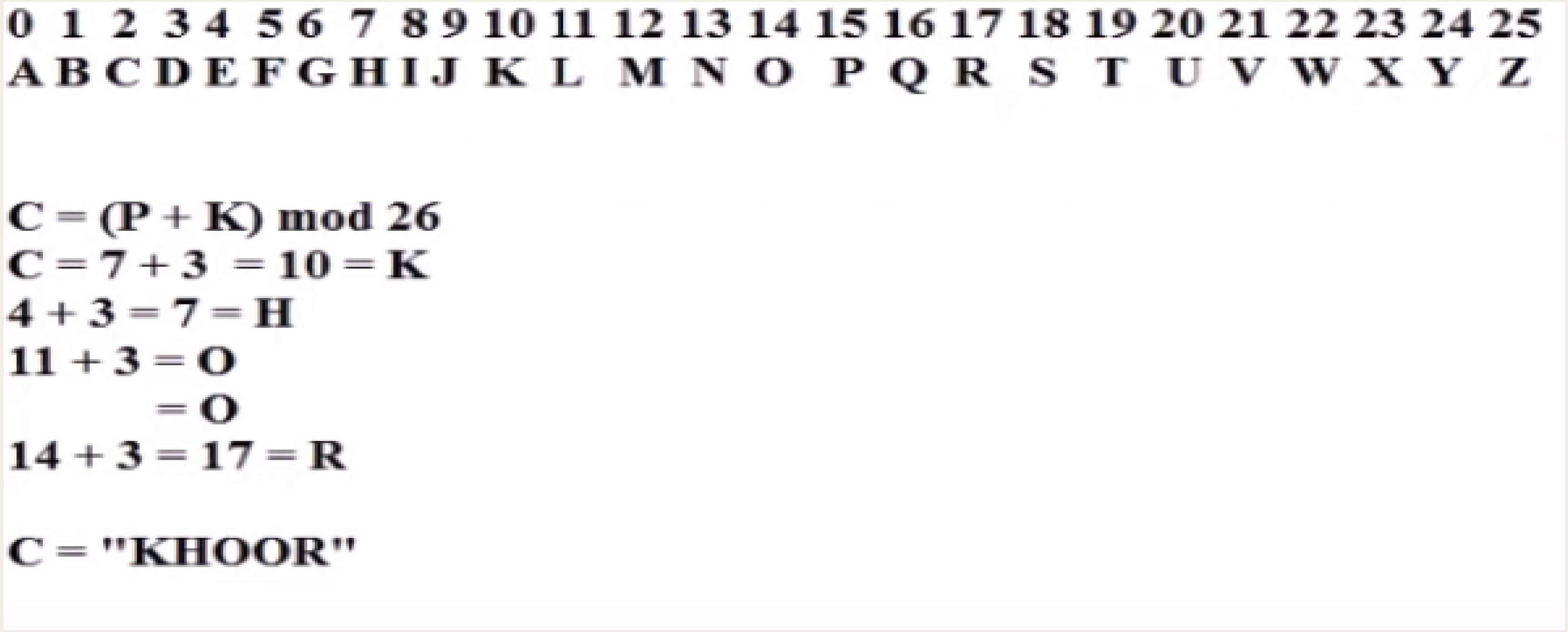
- Where k takes on a value in the range 1 to 25.
- The decryption algorithm is simply

$$p = \mathrm{D}(k, C) = (C - k) \bmod 26$$

Example:



Encrypt the message P="Hello" using Caesar Cipher, given the key K=3.



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
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

C = (P + K) mod 26

C = 7 + 3 = 10 = K

4 + 3 = 7 = H

11 + 3 = O

= O

14 + 3 = 17 = R

C = "KHOOR"

Python chr() and ord()

Python's built-in function **chr()** is used for converting an *Integer* to a *Character*, while the function **ord()** is used to do the reverse, i.e., convert a *Character* to an *Integer*.

```
# Convert integer 65 to ASCII Character ('A')
print(chr(65))

# Convert ASCII Unicode Character 'A' to 65
print(ord('A'))
```

Output:

```
    A
    65
```

Python isupper() and islower() Function

Python String **isupper()** function checks if all the characters in a string are uppercase then returns true else false

```
str_name = "WELCOME"
print(str_name.isupper())    # True
```

Python String **islower()** function checks if all the characters in a string are lowercase then return True else False.

```
str_name = "welcome"
print(str_name.islower())    # True
```

Python String lower() Method

Python String lower() method converts a string object into a lowercase string

```
a = 'WELCOME TO PYTHON'
print(a.lower())    # welcome to python
```

Example

The program implementation of Caesar cipher algorithm is as follows:

```python
text = 'Python'
k = 1
result =''
for i in range(len(text)):
    char = text[i]
    if (char.isupper()):
        # Encrypt uppercase characters
        s = chr((ord(char) - 65 + k) % 26 + 65)
    else:
        # Encrypt lowercase characters
        s = chr((ord(char) - 97 + k) % 26 + 97)
    result += s
print(result)
```

Output:

```
Qzuipo
```

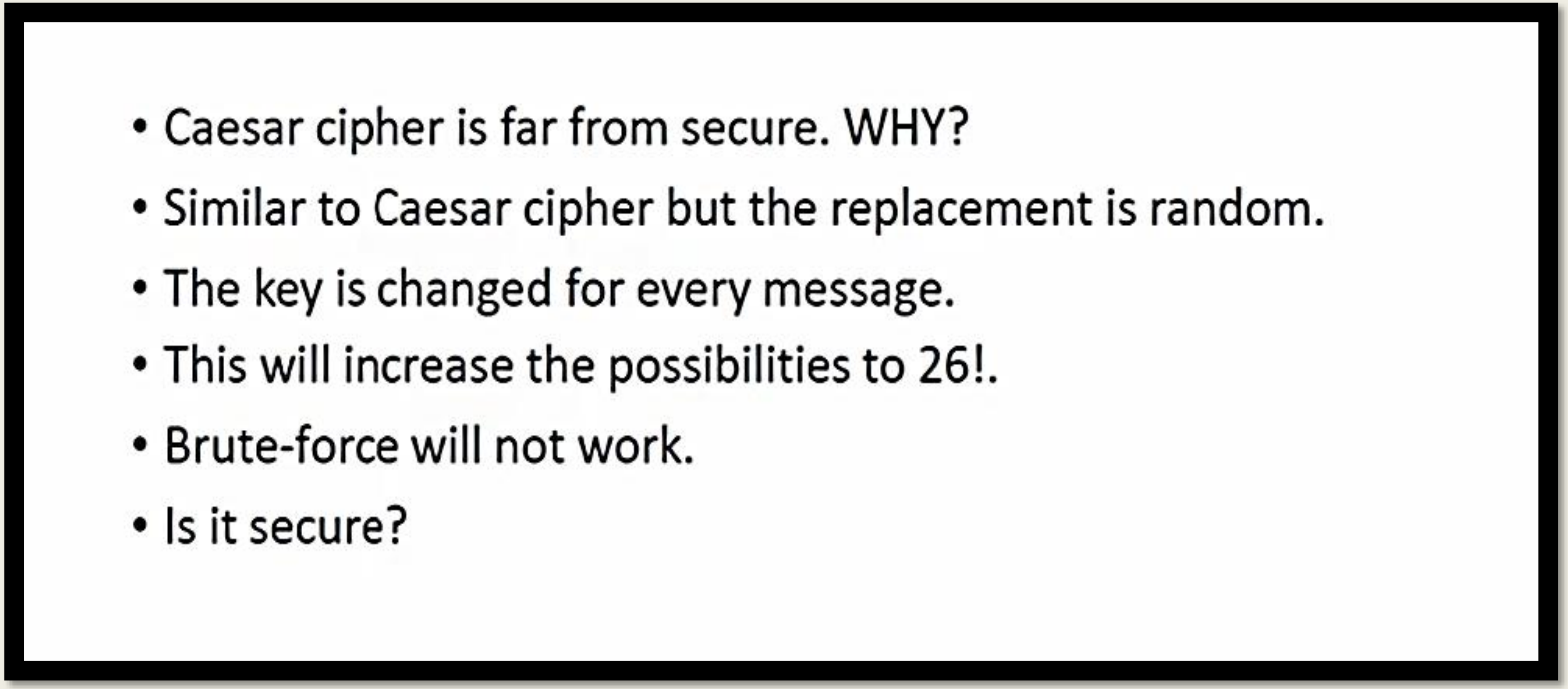
Caesar Cipher Decryption

```
text = 'Qzuipo'
k = 1
result =''
for i in range(len(text)):
    char = text[i]
    if (char.isupper()):
        # Encrypt uppercase characters
        s = chr((ord(char) - 65 - k) % 26 + 65)
    else:
        # Encrypt lowercase characters
        s = chr((ord(char) - 97 - k) % 26 + 97)
    result += s
print(result)
```

Output:

```
Python
```

Monoalphabetic Cipher



- Caesar cipher is far from secure. WHY?
- Similar to Caesar cipher but the replacement is random.
- The key is changed for every message.
- This will increase the possibilities to 26!.
- Brute-force will not work.
- Is it secure?

- Monoalphabetic work by replacing each letter of the plaintext with another letter (or possibly even a random symbol).
- A monoalphabetic substitution cipher, also known as a simple substitution cipher, relies on a fixed replacement structure. That is, the substitution is fixed for each letter of the alphabet. Thus, if "A" is encrypted to "R", then every time we see the letter "A" in the plaintext, we replace it with the letter "R" in the ciphertext.
- Example:

Alphabetic :	**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**
key :	**q**	**w**	**e**	**r**	**t**	**y**	**u**	**i**	**o**	**p**	**a**	**s**	**d**	**m**	**g**	**h**	**j**	**k**	**l**	**z**	**x**	**c**	**v**	**b**	**n**	**f**

- Encrypt the message "hello" using Monoalphabetic Cipher given the key above.

Plaintext = hello.

Ciphertext = itssg.

Create a list of alphabets

```
import string
# lowercase letters
print(string.ascii_lowercase)    #abcdefghijklmnopqrstuvwxyz

# uppercase letters
print(string.ascii_uppercase)    #ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

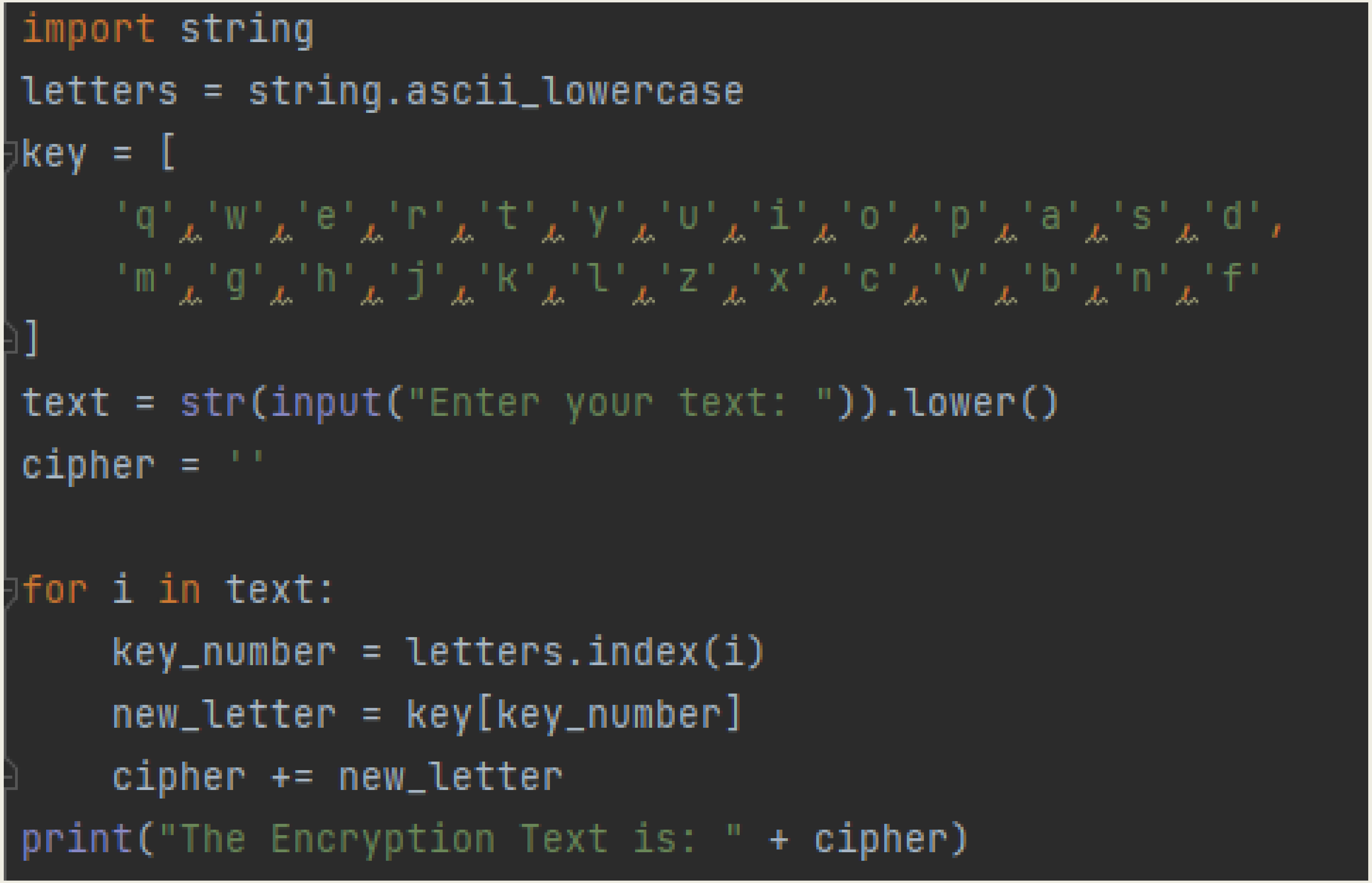
Monoalphabetic Cipher Encryption

```
letters = [
    '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']
key = [
    'q','w','e','r','t','y','u','i','o','p','a','s','d',
    'm','g','h','j','k','l','z','x','c','v','b','n','f'
]
text = str(input("Enter your text: ")).lower()
cipher = ''

for i in text:
    key_number = letters.index(i)
    new_letter = key[key_number]
    cipher += new_letter
print("The Encryption Text is: " + cipher)
```

Output:

```
Enter your text: Hello
The Encryption Text is: itssg
```



```
import string
letters = string.ascii_lowercase
key = [
    'q','w','e','r','t','y','u','i','o','p','a','s','d',
    'm','g','h','j','k','l','z','x','c','v','b','n','f'
]
text = str(input("Enter your text: ")).lower()
cipher = ''


for i in text:
    key_number = letters.index(i)
    new_letter = key[key_number]
    cipher += new_letter
print("The Encryption Text is: " + cipher)
```

Monoalphabetic Cipher Decryption

```python
letters = [
    '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']
key = [
    'q','w','e','r','t','y','u','i','o','p','a','s','d',
    'm','g','h','j','k','l','z','x','c','v','b','n','f'
]
text = str(input("Enter your text: ")).lower()
#Encryption
cipher = ''
for i in text:
    key_number = letters.index(i)
    new_letter = key[key_number]
    cipher += new_letter
print("The Encryption Text is: " + cipher)
# Decryption
plaintext = ''
for c in cipher:
    key_number2 = key.index(c)
    new_plain = letters[key_number2]
    plaintext += new_plain
print("The Decryption Text is: " + plaintext)
```

Output:

```
Enter your text: Hello
The Encryption Text is: itssg
The Decryption Text is: hello
```

Task

- Encrypt the message = “welcome” using Playfair Cipher.

Thank You