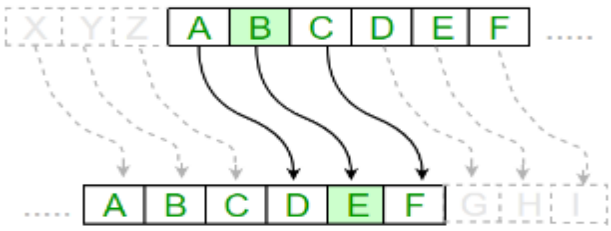
Cryptography with Python - Caesar Cipher

1. 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.
- Replace each letter by the letter **three positions** along in alphabet.

```
Plain : 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
Cipher: 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
```



- Allow shift by k positions.
- It representing each letter as a number called an ordinal, and then adding or subtracting from this number to form a new ordinal (and a new letter).

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

Where C: Cipher Text

P: Plain Text

E: Encryption algorithm

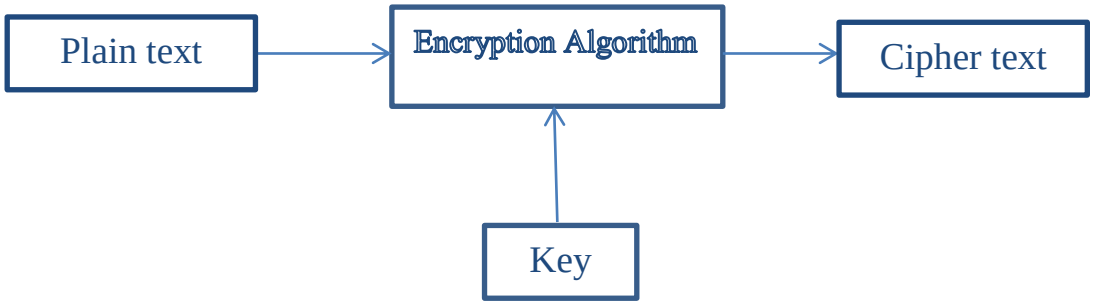
D: Decryption algorithm

K: Key

- ASCII is a code that connects each character to a number between 32 and 126. The capital letters "A" through "Z" have the ASCII numbers 65 through 90. The lowercase letters "a" through "z" have the ASCII numbers 97 through 122. The numeric digits "0" through "9" have the ASCII numbers 48 through 57. The following table shows all the ASCII characters and ordinals.

32	(space)	48	0	64	@	80	P	96	`	112	p
33	!	49	1	65	A	81	Q	97	a	113	q
34	"	50	2	66	B	82	R	98	b	114	r
35	#	51	3	67	C	83	S	99	c	115	s
36	$	52	4	68	D	84	T	100	d	116	t
37	%	53	5	69	E	85	U	101	e	117	u
38	&	54	6	70	F	86	V	102	f	118	v
39	'	55	7	71	G	87	W	103	g	119	w
40	(	56	8	72	H	88	X	104	h	120	x
41	)	57	9	73	I	89	Y	105	i	121	y
42	*	58	:	74	J	90	Z	106	j	122	z
43	+	59	;	75	K	91	[	107	k	123	{
44	,	60	<	76	L	92	\	108	l	124	|
45	-	61	=	77	M	93	]	109	m	125	}
46	.	62	>	78	N	94	^	110	n	126	~
47	/	63	?	79	O	95	_	111	o		

1.1 Encryption Phase



- ✓ **Input : Plain text, Key**
- ✓ **Output: Cipher text**
- ✓ **Encryption Algorithm = (P + K) mod 26**

To implement the Caesar encryption algorithm, follow these steps:

Step 1) Taking input in Python: function to read the input from the keyboard.

In Python 2 to accept user input we can use the following two functions:-

- **input( )**
- **raw_input( )**

The main difference between those two functions is input() function automatically converts user input to appropriate type. i.e., If a user-entered string input() function converts it into a string, and if a user entered a number it converts to an integer. While the raw_input() function convert every user input to string.

raw_input() function of python 2 is renamed to input() in Python 3.x and original input() function is removed from Python 3.

input() function always converts input into a string. If you enter an integer value, still input() function converts it into a string. We need to convert an input value into an integer type explicitly.

For example:-

```
1   # Python program showing
2   # a use of input()
3
4   PlainText = input("Enter The PlainText: ")
5   print(PlainText)
6   print(type(PlainText))
7
8   Key = int(input("Enter Key value: "))
9   print(Key)
10  print(type(Key))
```

Result:-

```
Enter The PlainText: hello
hello
<class 'str'>
Enter Key value: 3
3
<class 'int'>
```

Step 2) Encryption code:

```
def Encrypt(PlainText, Key):
    CipherText = ""
# transverse the plain text
    for char in PlainText:
        # Encrypt uppercase characters in plain text
        if (char.isupper()):
            CipherText += chr((ord(char) + Key - 65) % 26 + 65)

        # Encrypt lowercase characters in plain text
        else:
            CipherText += chr((ord(char) + Key - 97) % 26 + 97)
    return CipherText
```

- **The chr() function** (pronounced "char", short for "character") takes an integer ordinal and returns a single-character string. **chr(97) ➔ a**
- **The ord() function** (short for "ordinal") takes a single-character string, and returns the integer ordinal value. **ord('a') ➔ 97**

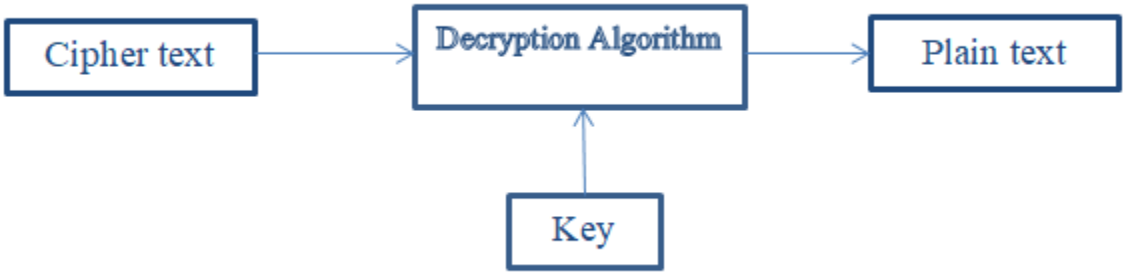
Step 3) Call Function:

```
# call the above function
PlainText = input("Enter The PlainText: ")
Key = int(input("Enter Key Value: "))

CipherText = Encrypt(PlainText, Key)

print("Plain Text : " + PlainText)
print("Cipher: " + CipherText)
```

Result:-

```
Enter The PlainText: HelloSir
Enter Key Value: 3
Plain Text : HelloSir
Cipher: KhoorVlu
```

1.2 Decryption Phase



- ✓ **Input : Cipher text, Key**
- ✓ **Output: Plain text**
- ✓ **Decryption Algorithm = (C - K) mod 26**

Decryption Code:

```
def Decrypt(CipherText, Key):
    PlainText = ""
# transverse the cipher text
    for char in CipherText:
        # Decrypt uppercase characters in Cipher text
        if (char.isupper()):
            PlainText += chr((ord(char) - Key - 65) % 26 + 65)

        # Decrypt lowercase characters in cipher text
        else:
            PlainText += chr((ord(char) - Key - 97) % 26 + 97)
    return PlainText
```