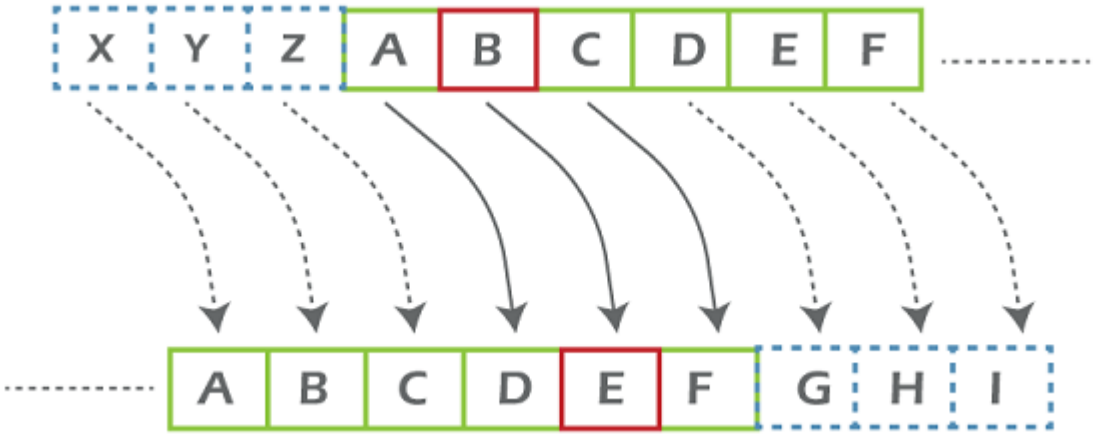
Caesar Cipher Program in Javascript & Python

It is one of the simplest and most used encryption techniques. In this technique, each letter of the given text is replaced by a letter of some fixed number of positions down the alphabet.

For example, with a shift of 1, X would be replaced by Y, Y would become Z, and so on. Julius Caesar was the first one who used it for communicating with his officials. Based on his name, this technique was named as Caesar Cipher technique.

An integer value is required to cipher a given text. The integer value is known as shift, which indicates the number of positions each letter of the text has been moved down.



We can mathematically represent the encryption of a letter by a shift n in the following way:

Encryption phase with shift n = $E_n(x) = (x+n) \bmod 26$

Decryption phase with shift n = $D_n(x) = (x-n) \bmod 26$

Examples:
Text: ABCDEFGHIJKLMNOPQRSTUVWXYZ

Shift: 23

Cipher: XYZABCDEFGHIJKLMNOPQRSTUVW

Text: ATTACKATONCE

Shift: 4

Cipher: EXXEGOEXSRGI

Text: DIGITALREGENESYS

Shift: 8

Cipher:

Text: PYTHONPROGRAMMING

Shift: 6

Cipher:

Experiment 11:

To encrypt and decrypt the given message by using Ceaser Cipher encryption algorithm.

ALGORITHMS:

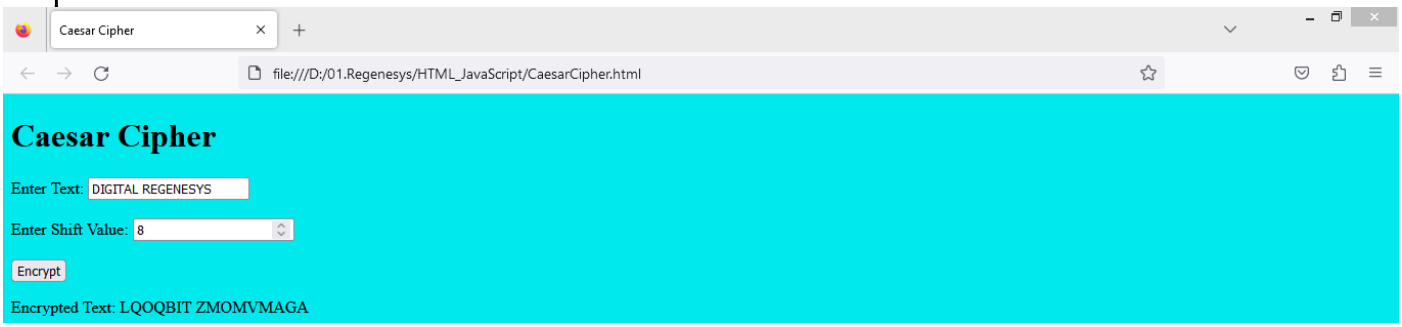
1. In Ceaser Cipher each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet.
2. For example, with a left shift of 3, D would be replaced by A, E would become B, and so on.
3. The encryption can also be represented using modular arithmetic by first transforming the letters into numbers, according to the scheme, A = 0, B = 1, Z = 25.
4. Encryption of a letter x by a shift n can be described mathematically as,
5. $E_n(x) = (x + n) \mod 26$
6. Decryption is performed similarly,
7. $D_n(x) = (x - n) \mod 26$

11.1. HTML_JAVA SCRIPT PROGRAM:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Caesar Cipher</title>
</head>
<body bgcolor=Unbleached Silk>
    <h1>Caesar Cipher</h1>
    <label for="inputText">Enter Text:</label>
    <input type="text" id="inputText"><br/><br/>

    <label for="shiftValue">Enter Shift Value:</label>
    <input type="number" id="shiftValue"><br/><br/>

    <button onclick="encryptText()">Encrypt</button>
    <p id="result"></p>

    <script>
        function caesarCipher(text, shift)
    {
            let result = "";
            for (let i = 0; i < text.length; i++)
        {
                let char = text.charAt(i);

                if (char.match(/[a-z]/i))
        {
                    let isUpperCase = char === char.toUpperCase();
                    let shiftedChar = String.fromCharCode((char.charCodeAt(0) + shift -
(isUpperCase ? 65 : 97)) % 26 + (isUpperCase ? 65 : 97));
                    result += shiftedChar;
                }
        else
        {
                    result += char;
                }
            }
            return result;
```

```
        }

        function encryptText()
    {
            let inputText = document.getElementById('inputText').value;
            let shiftValue = parseInt(document.getElementById('shiftValue').value);

            if (!isNaN(shiftValue))
        {
                let encryptedText = caesarCipher(inputText, shiftValue);
                document.getElementById('result').innerText = `Encrypted Text:
${encryptedText}`;
            }
        else
           {
                document.getElementById('result').innerText = "Please enter a valid shift
value.";
            }
        }
    </script>
</body>
</html>
```

Output:



11.2. PYTHON PROGRAM:

```
# A python program to illustrate Caesar Cipher Technique
def encrypt(text, s):
    result = ""

    # traverse text
    for i in range(len(text)):
        char = text[i]

        # Encrypt uppercase characters
        if (char.isupper()):
            result += chr((ord(char) + s - 65) % 26 + 65)

        # Encrypt lowercase characters
        else:
            result += chr((ord(char) + s - 97) % 26 + 97)

    return result
```

```
# check the above function
text = "DIGITAL REGENESYS"
s = 8
print("Text  : " + text)
print("Shift : " + str(s))
print("Cipher: " + encrypt(text, s))
```

Output:

```
Text  : DIGITAL REGENESYS
Shift : 8
Cipher: LQOQBIT ZMOMVMAGA
```

11.3. PYTHON PROGRAM:

```
def caesar_cipher(text, shift):
    result = ""

    for char in text:
        if char.isalpha():
            # Determine the case (upper or lower)
            is_upper = char.isupper()
            # Apply the shift and ensure it stays within the alphabet
            shifted_char = chr((ord(char) + shift - ord('A' if is_upper else 'a')) % 26 +
ord('A' if is_upper else 'a'))
            result += shifted_char
        else:
            result += char

    return result

# Accept input from the user
text_to_encrypt = input("Enter the text to encrypt: ")
shift_value = int(input("Enter the shift value: "))

# Encrypt the text
encrypted_text = caesar_cipher(text_to_encrypt, shift_value)

# Display the results
print(f"Original text: {text_to_encrypt}")
print(f"Encrypted text: {encrypted_text}")
```

Output:

```
Enter the text to encrypt: DIGITAL REGENESYS
Enter the shift value: 8
Original text: DIGITAL REGENESYS
Encrypted text: LQOQBIT ZMOMVMAGA
```

NOTE:
In the Python code for the Caesar Cipher, the ord() function is used to obtain the ASCII (American Standard Code for Information Interchange) value of a character. The name "ord" stands for "ordinal," and the function returns the integer representing the Unicode code point of the given character.

In the context of the Caesar Cipher program, ord(char) returns the ASCII value of the character char. This value is then used in arithmetic operations to perform the shift in the alphabet.

For example:

ord('A') returns 65
ord('a') returns 97
These values are essential for determining the position of the character in the alphabet, and they are used in the calculation of the shifted character in the Caesar Cipher algorithm.

The ASCII Table

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		