

# Experiment No. 01

## PART- II Elective-II- Information Security

### Title

Write a Java/C/C++/Python program that contains a string (char pointer) with a value 'HelloWorld'. The program should AND and XOR each character in this string with 127 and display the result.

### Objective

Implement AND, OR and XOR each character with 127

### Outcome

Student should be able to see the result of AND, OR and XOR each character with 127.

### Theory

#### String

The string is the one-dimensional array of characters terminated by a null ('\0').

Each and every character in the array consumes one byte of memory, and the last character must always be '\0'.

The termination character ('\0') is used to identify where the string ends.

In C language string declaration can be done in two ways

- 1.By char array
- 2.By string literal

#### 1.By char array

```
char ch[17]={ 'o', 'n', 'i', 'n', 'e', 's', 'm', 'a', 'r', 't', 't', 'r', 'a', 'i', 'n', 'e', 'r', '\0' };
```

As we know, array index starts from 0, so it will be represented as in the figure given below.

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Character	o	n	i	n	e	s	m	a	r	t	t	r	a	i	n	e	r	\0	

While declaring string, size is not mandatory. So we can write the above code as given below: `char ch[]={ 'o', 'n', 'i', 'n', 'e', 's', 'm', 'a', 'r', 't', 't', 'r', 'a', 'i', 'n', 'e', 'r', '\0' }`

#### 2.By string literal

We can also define the string by the string literal in C language. For example:

```
char str[]="onlinesmarttrainer";
```

In such case, '\0' will be appended at the end of the string by the compiler.

## AND Operation

There are two inputs and one output in binary AND operation.

The inputs and result to a binary AND operation can only be 0 or 1.

The binary AND operation will always produce a 1 output if both inputs are 1 and will produce a 0 output if both inputs are 0. For two different inputs, the output will be 0.

Symbol	Truth Table		
 2-input AND Gate	A	B	Q
	0	0	0
	0	1	0
	1	0	0
	1	1	1
Boolean Expression $Q = A \cdot B, A \text{ OR } B$			

Fig: AND Gate

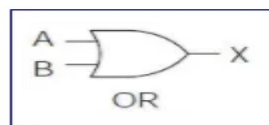
## OR Operation

There are two inputs and one output in binary OR operation.

The inputs and result to a binary OR operation can only be 0 or 1.

The OR gate is a mostly used digital logic circuit. The output state of the OR gate will always be low when both of the inputs states is low.

Simply, if any input value in the OR gate is set to 1, then it will always return high-level output(1).



OR gate		
Input A	Input B	Output
0	0	0
1	0	1
0	1	1
1	1	1

Fig: OR Gate

## XOR Operation

There are two inputs and one output in binary **XOR** (exclusive **OR**) operation.

**It is** similar to **ADD** operation which takes two inputs and produces one result i.e. one output.

The inputs and result to a binary **XOR** operation can only be **0** or **1**.

The binary **XOR** operation will always produce a **1** output if either of its inputs is **1** and will produce a **0** output if both of its inputs are **0** or **1**.

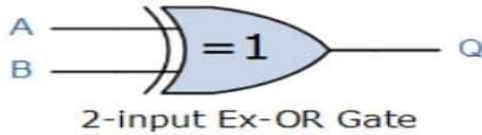
Symbol	Truth Table		
 <p>2-input Ex-OR Gate</p>	A	B	Q
	0	0	0
	0	1	1
	1	0	1
	1	1	0
Boolean Expression $Q = A \text{ XOR } B$			

Fig: XOR Gate

### Algorithm

1. Start
2. Take the input 'hello world' which is assigned to variable 'str'
3. Perform AND operation between the string and 127.
4. Perform OR operation between the string and 127
5. Perform XOR operation between the string and 127
6. Then print the result
7. Stop.

### Conclusion

Thus we have studied AND, OR and XOR gate operation on string to identify the encryption process.