**Assignment No 1**

**Title:**

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

Theory:

For this assignment we need to know java logical operations and String class methods

**Logical Operations:**

Logical operators are used to performing logical “AND”, “OR” and “NOT” operations, i.e. the function similar to AND gate and OR gate in digital electronics. They are used to combine two or more conditions/constraints or to complement the evaluation of the original condition under particular consideration. One thing to keep in mind is, while using AND operator, the second condition is not evaluated if the first one is false. Whereas while using OR operator, the second condition is not evaluated if the first one is true, i.e. the AND and OR operators have a short-circuiting effect. Used extensively to test for several conditions for making a decision.

1. AND Operator ( && ) – if( a && b ) [if true execute else don’t]
2. OR Operator ( || ) – if( a || b) [if one of them is true execute else don’t]
3. NOT Operator ( ! ) – !(a<b) [returns false if a is smaller than b]

**Example:**

import java.io.\*;

class Test {

    public static void main(String[] args)

    {

        int a = 12;

        int b = 25;

        System.out.println("Demonstrating & operator\n");

        int c = a & b;

        System.out.println(a + " & " + b + " = " + c);

    }

}

**Output:**

12 & 25 = 8

**String class and methods:**

Strings, which are widely used in Java programming, are a sequence of characters. In Java programming language, strings are treated as objects.

The Java platform provides the String class to create and manipulate strings.

The most direct way to create a string is to write −

String s1 = "Hello world!";

Whenever it encounters a string literal in your code, the compiler creates a String object with its value in this case, "Hello world!'.

**Syntax:**

<String\_Type> <string\_variable> = "<sequence\_of\_string>";

**e.g.** String str = "welcomes";

public class StringDemo {

    public static void main(String args[])

    {

        String s1 = new String("example");

        System.out.println(s1);

    }

}

Output: example

**String Methods:**

**char charAt(index i) method**

The charAt() method returns the character at the specified index in a string. The index of the first character is 0, the second character is 1, and so on.

**Syntax:**

public char charAt(int *index*)

here index is the int value representing index of the character to return.

Example:

Public class Demo{

public static void main(String args[])

{

String s1=”Good Morning….”;

char ch= s1.chatAt(2);

System.out.println(ch);

}}

**int length() method**

This method returns number of character in string

**Syntax:**

public int length()

it returns an int value, representing the length of the string.

Example:

Example:

Public class Demo{

public static void main(String args[])

{

String s1=”Good Morning”;

int len=s1.length();

System.out.println(len);

}}

Output:12

**Algorithm:**

This algorithm outlines the key steps of the program, providing a structured approach to understand the logic behind the bitwise operations on each character in the given string.

1. Initialize the string s with the value "Helloworld".
2. Iterate through each character in the string using a for loop with index i ranging from 0 to the length of the string minus 1.
3. Retrieve the ASCII value of the current character using s.charAt(i).
4. Apply bitwise XOR operation with 127 to the ASCII value:

**int a = (s.charAt(i) ^ 127);**

1. Convert the result of the XOR operation back to a character and store it in a **string: String str = new Character((char) a).toString();**
2. Display the original character and the result of the bitwise AND, OR, and XOR operations with 127.

* Bitwise AND: s.charAt(i) & 127
* Bitwise OR: s.charAt(i) | 127
* Bitwise XOR: s.charAt(i) ^ 127

1. Move to the next line for a new character.

**Conclusion:**

In conclusion, the provided Java/C/C++/Python program successfully demonstrated the bitwise AND and XOR operations on each character of the string "HelloWorld" with the value 127. These operations were applied to manipulate the bits of each character, resulting in a transformed output. The bitwise AND operation with 127 clears the higher-order bit, while the XOR operation toggles certain bits.