

Title :

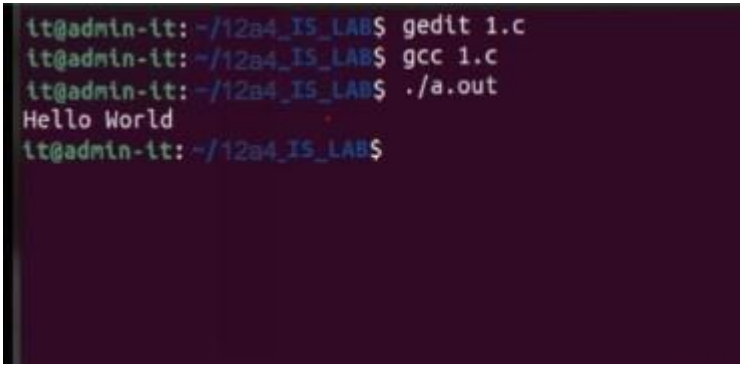
Date :

Page No. :

1. Write a C program that contains a string (char pointer) with a value 'Hello world'. The program should XOR each character in this string with 0 and displays the result.

```
#include <stdio.h>
#include <string.h>
int main() {
    char *str = "Hello world";
    int len = strlen(str);
    for (int i = 0; i < len; i++) {
        printf("%c", (str[i] ^ 0));
    }
    return 0;
}
```

Output:



```
it@admin-it: ~/12a4_IS_LAB$ gedit 1.c
it@admin-it: ~/12a4_IS_LAB$ gcc 1.c
it@admin-it: ~/12a4_IS_LAB$ ./a.out
Hello World
it@admin-it: ~/12a4_IS_LAB$
```

Title :

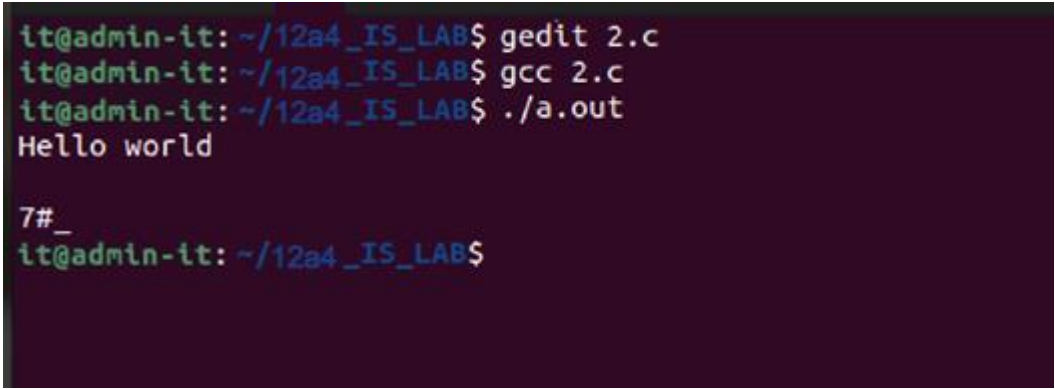
Date :

Page No. :

2. Write a C program that contains a string (char pointer) with a value 'Hello world'. The program should AND, OR and XOR each character in this string with 127 and display the result.

```
#include <stdio.h>
#include <string.h>
int main() {
    char *str = "Hello world";
    int len = strlen(str);
    for (int i = 0; i < len; i++) {
        printf("%c", (str[i] & 127));
    }
    printf("\n");
    for (int i = 0; i < len; i++) {
        printf("%c", (str[i] | 127));
    }
    printf("\n");
    for (int i = 0; i < len; i++) {
        printf("%c", (str[i] ^ 127));
    }
    return 0;
}
```

Output:



```
it@admin-it: ~/12a4_IS_LAB$ gedit 2.c
it@admin-it: ~/12a4_IS_LAB$ gcc 2.c
it@admin-it: ~/12a4_IS_LAB$ ./a.out
Hello world

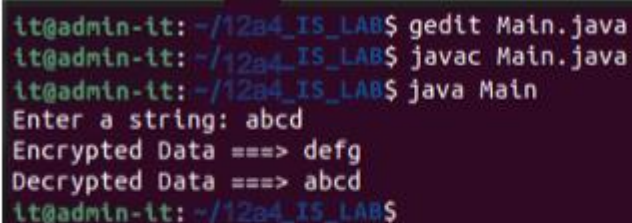
7#_
it@admin-it: ~/12a4_IS_LAB$
```

3. Write a Java program to perform encryption and decryption using the following algorithms**a. Caesar cipher**

```
import java.util.Scanner;
public class Main {
    public static final String ALPHABET = "abcdefghijklmnopqrstuvwxyz";

    public static String processData(String inputStr, int shiftKey, boolean encrypt) {
        inputStr = inputStr.toLowerCase();
        StringBuilder result = new StringBuilder();
        for (int i = 0; i < inputStr.length(); i++) {
            int pos = ALPHABET.indexOf(inputStr.charAt(i));
            int shift = encrypt ? shiftKey : -shiftKey;
            int newPos = (pos + shift + 26) % 26;
            result.append(ALPHABET.charAt(newPos));
        }
        return result.toString();
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String inputStr = sc.nextLine();
        int shiftKey = 3;
        String encrypted = processData(inputStr, shiftKey, true);
        System.out.println("Encrypted Data ==> " + encrypted);
        System.out.println("Decrypted Data ==> " + processData(encrypted, shiftKey, false));
        sc.close();
    }
}
```

Output:

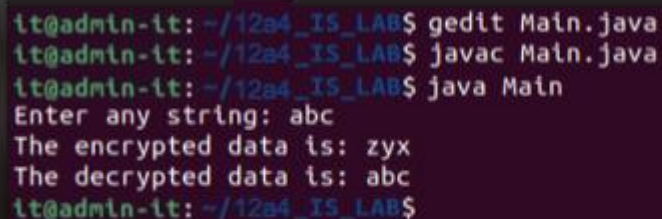
```
it@admin-it: ~/12a4_IS_LAB$ gedit Main.java
it@admin-it: ~/12a4_IS_LAB$ javac Main.java
it@admin-it: ~/12a4_IS_LAB$ java Main
Enter a string: abcd
Encrypted Data ==> defg
Decrypted Data ==> abcd
it@admin-it: ~/12a4_IS_LAB$
```

b) Substitution cipher

```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String a = "abcdefghijklmnopqrstuvwxyz", b = "zyxwvutsrqponmlkjihgfedcba";
        System.out.print("Enter any string: ");
        String str = sc.nextLine();
        StringBuilder encrypted = new StringBuilder(), decrypted = new StringBuilder();

        // Encryption
        for (char c : str.toCharArray()) {
            int j = a.indexOf(c);
            encrypted.append(j != -1 ? b.charAt(j) : c);
        }
        // Decryption
        for (char c : encrypted.toString().toCharArray()) {
            int j = b.indexOf(c);
            decrypted.append(j != -1 ? a.charAt(j) : c);
        }

        System.out.println("Encrypted data: " + encrypted);
        System.out.println("Decrypted data: " + decrypted);
        sc.close();
    }
}
```

Output:

```
it@admin-1t: ~/12a4_IS_LAB$ gedit Main.java
it@admin-1t: ~/12a4_IS_LAB$ javac Main.java
it@admin-1t: ~/12a4_IS_LAB$ java Main
Enter any string: abc
The encrypted data is: zyx
The decrypted data is: abc
it@admin-1t: ~/12a4_IS_LAB$
```

c) Hill cipher

```
import java.util.Scanner;
public class Main {
    static Scanner sc = new Scanner(System.in);
    public static void main(String args[]) {
        System.out.print("Enter n: ");
        int n = sc.nextInt();
        int pt[][] = new int[1][n];
        System.out.print("Enter plaintext matrix [1 x n]: ");
        for(int i = 0; i < n; i++) {
            char x = sc.next().charAt(0);
            pt[0][i] = x - 'A';
        }
        int key[][] = new int[n][n];
        System.out.println("Enter key matrix [n x n]: ");
        for(int i = 0; i < n; i++) {
            for(int j = 0; j < n; j++) {
                key[i][j] = sc.nextInt();
            }
        }
        System.out.println("Encrypted text:");
        matrixMultiply(pt, key, n, true);
        sc.close();
    }

    public static void matrixMultiply(int[][] mat1, int[][] mat2, int n, boolean flag) {
        int[][] resultMat = new int[1][n];
        for (int i = 0; i < 1; i++) {
            for (int j = 0; j < n; j++) {
                resultMat[i][j] = 0;
                for (int k = 0; k < n; k++) {
                    resultMat[i][j] += mat1[i][k] * mat2[k][j];
                }
            }
        }
        for(int i = 0; i < n; i++) {
            char x = (char)((resultMat[0][i] % 26) + 'A');
            System.out.print(x + " ");
        }

        if(flag) {
            int inverse[][] = new int[n][n];
            System.out.println("\nEnter inverse key matrix [n x n]: ");
            for(int i = 0; i < n; i++) {
                for(int j = 0; j < n; j++) {
                    inverse[i][j] = sc.nextInt();
                }
            }
        }
    }
}
```

Title :

Date :

Page No. :

```
    }  
    System.out.println("Decrypted text:");  
    matrixMultiply(resultMat, inverse, n, false);  
  }  
}  
}
```

Output:



```
lt@admin-lt: ~/12th_15_LAN$ gedit Main.java  
lt@admin-lt: ~/12th_15_LAN$ javac Main.java  
lt@admin-lt: ~/12th_15_LAN$ java Main  
Enter n: 3  
Enter plaintext matrix [1 x n]: P A Y  
Enter key matrix [n x n]:  
17 17 5  
21 18 21  
2 2 19  
Encrypted text:  
R R L  
Enter inverse key matrix [n x n]:  
4 9 15  
15 17 6  
24 0 17  
Decrypted text:  
P A Y  
lt@admin-lt: ~/12th_15_LAN$
```

4. Write a C/JAVA program to implement the DES algorithm logic.

```
import javax.crypto.Cipher;
import javax.crypto.spec.SecretKeySpec;
import java.util.*;

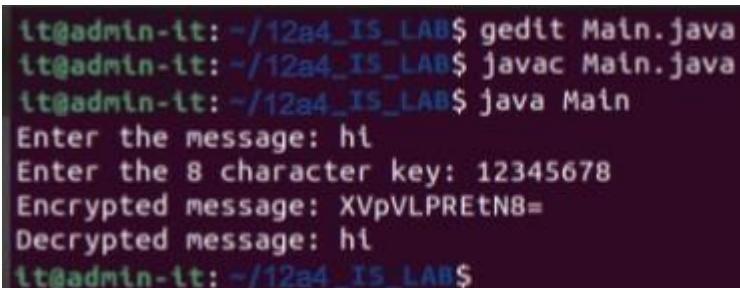
public class Main {
    public static void main(String[] args) throws Exception {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the message: ");
        String message = sc.nextLine();
        System.out.print("Enter the 8 character key: ");
        String key = sc.nextLine();

        byte[] encryptedMessage = encrypt(message, key);
        System.out.println("Encrypted message: " +
Base64.getEncoder().encodeToString(encryptedMessage));
        System.out.println("Decrypted message: " + decrypt(encryptedMessage, key));
    }

    public static byte[] encrypt(String message, String key) throws Exception {
        return cipher(message.getBytes(), key, Cipher.ENCRYPT_MODE);
    }

    public static String decrypt(byte[] encryptedMessage, String key) throws Exception {
        return new String(cipher(encryptedMessage, key, Cipher.DECRYPT_MODE));
    }

    public static byte[] cipher(byte[] input, String key, int mode) throws Exception {
        Cipher cipher = Cipher.getInstance("DES/ECB/PKCS5Padding");
        cipher.init(mode, new SecretKeySpec(key.getBytes(), "DES"));
        return cipher.doFinal(input);
    }
}
```

Output:

```
lt@admin-1t: ~/12a4_IS_LAB$ gedit Main.java
lt@admin-1t: ~/12a4_IS_LAB$ javac Main.java
lt@admin-1t: ~/12a4_IS_LAB$ java Main
Enter the message: hl
Enter the 8 character key: 12345678
Encrypted message: XVpVLPREtN8=
Decrypted message: hl
lt@admin-1t: ~/12a4_IS_LAB$
```

5. Write a C/JAVA program to implement the Blowfish algorithm logic.

```
import javax.crypto.Cipher;
import javax.crypto.spec.SecretKeySpec;
import java.util.*;

public class Main {
    public static String encrypt(String plainText, String key) throws Exception {
        Cipher cipher = Cipher.getInstance("Blowfish");
        cipher.init(Cipher.ENCRYPT_MODE, new SecretKeySpec(key.getBytes(), "Blowfish"));
        return Base64.getEncoder().encodeToString(cipher.doFinal(plainText.getBytes()));
    }

    public static String decrypt(String cipherText, String key) throws Exception {
        Cipher cipher = Cipher.getInstance("Blowfish");
        cipher.init(Cipher.DECRYPT_MODE, new SecretKeySpec(key.getBytes(), "Blowfish"));
        return new String(cipher.doFinal(Base64.getDecoder().decode(cipherText)));
    }

    public static void main(String[] args) throws Exception {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter plain text: ");
        String password = sc.nextLine();
        System.out.print("Enter Key: ");
        String key = sc.nextLine();
        String enc_output = encrypt(password, key);
        System.out.println("Encrypted text: " + enc_output);
        System.out.println("Decrypted text: " + decrypt(enc_output, key));
    }
}
```

Output:

```
it@admin-it: ~/12a4_IS_LAB$ gedit Main.java
it@admin-it: ~/12a4_IS_LAB$ javac Main.java
it@admin-it: ~/12a4_IS_LAB$ java Main
Enter plain text: Hello World
Enter Key: Hello
Encrypted text: L8N/CwZv9a6r/42LAlyfHw==
Decrypted text: Hello World
it@admin-it: ~/12a4_IS_LAB$
```