1. a) Implementation of the Data Link Layer Framing methods Bit stuffing.

**Aim : Program to Implementation of the Data Link Layer Framing methods Bit stuffing**

Program :

#include <stdio.h>

#include <string.h>

#define MAX\_SIZE 100

// Function to perform bit stuffing

void bitStuffing(char input[], char output[]) {

int i, j = 0, count = 0;

int length = strlen(input);

for (i = 0; i < length; i++) {

// Copy the current bit to the output

output[j] = input[i];

j++;

// If the current bit is '1', increase the count

if (input[i] == '1') {

count++;

} else {

// If the current bit is '0', reset the count

count = 0;

}

// If there are 5 consecutive '1's, insert a '0'

if (count == 5) {

output[j] = '0';

j++;

count = 0;

}

}

// Null terminate the output string

output[j] = '\0';

}

int main() {

char input[MAX\_SIZE], output[MAX\_SIZE];

printf("Enter the input bit string: ");

scanf("%s", input);

bitStuffing(input, output);

printf("Bit-stuffed output: %s\n", output);

return 0;

}

Output :

Enter the input bit string: 11111100111111

Bit-stuffed output: 1111101001111101

**1.b) Implementation of the Data Link Layer Framing methods Character Stuffing .**

**Aim : Program to Implementation of the Data Link Layer Framing methods Character Stuffing**

#include <stdio.h>

#include <string.h>

#define MAX\_SIZE 100

void charStuffing(char input[], char output[])

{

int i, j = 0;

int length = strlen(input);

char flag = 'F'; // Flag character

char escape = 'E'; // Escape character

// Add the starting flag to the output

output[j++] = flag;

for (i = 0; i < length; i++) {

// If the current character is a flag or escape, add an escape character

if (input[i] == flag || input[i] == escape) {

output[j++] = escape;

}

// Copy the current character to the output

output[j++] = input[i];

}

// Add the ending flag to the output

output[j++] = flag;

// Null terminate the output string

output[j] = '\0';

}

int main() {

char input[MAX\_SIZE], output[MAX\_SIZE];

printf("Enter the input character string: ");

scanf("%s", input);

charStuffing(input, output);

printf("Character-stuffed output: %s\n", output);

return 0;

}

**OUTPUT**

**Enter the input character string: SVCET**

**Character-stuffed output: FSVCEETF**