BIT STUFFING PROGRAM

#include <stdio.h>

#include <string.h>

#define FLAG 0x7E // Flag sequence: 01111110

void bit\_stuffing(char \*input, char \*output) {

int i, j, count;

// Copy flag sequence at the beginning of the output

output[0] = FLAG;

j = 1;

count = 0;

// Perform bit stuffing

for (i = 0; i < strlen(input); i++) {

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

count++;

output[j] = '1';

} else {

count = 0;

output[j] = '0';

}

j++;

if (count == 5) {

// Stuff '0' after 5 consecutive '1's

output[j] = '0';

j++;

count = 0;

}

}

// Copy flag sequence at the end of the output

output[j] = FLAG;

}

void bit\_destuffing(char \*input, char \*output) {

int i, j, count;

// Skip flag sequences

i = 1;

j = 0;

count = 0;

// Perform bit destuffing

while (i < strlen(input) - 1) {

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

count++;

output[j] = '1';

} else {

count = 0;

output[j] = '0';

}

i++;

j++;

if (count == 5 && input[i] == '0') {

// Skip stuffed '0'

i++;

count = 0;

}

}

output[j] = '\0'; // Null-terminate the output string

}

int main() {

char input[] = "0111110101010101111110"; // Example input data with flag sequence at the beginning and end

char output[100]; // Output buffer

char destuffed\_output[100]; // Destuffed output buffer

printf("Input data: %s\n", input);

// Perform bit stuffing

bit\_stuffing(input, output);

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

// Perform bit destuffing

bit\_destuffing(output, destuffed\_output);

printf("Bit-destuffed data: %s\n", destuffed\_output);

return 0;

}

