**Computer Networks**

**Assignment 5**

**Name:** Gargi Dandare **Batch-2 IT-A**

**Roll No. :** 33

**Problem Statement**:

**IMPLEMENT FOLLOWING FRAMING METHODS**

**1.** **BYTE COUNT**

**2.** **FLAG BYTE WITH BYTE STUFFING**

**3. FLAG BYTE WITH BIT STUFFING**

**Code**: **BYTE COUNT**

#include <stdio.h>

#include <stdint.h>

#define MAX\_MESSAGE\_SIZE 1024

// Function to concatenate frames

void concatenateFrames(uint8\_t \*destination, const uint8\_t \*source, size\_t sourceLength) {

    // Find the NULL character in the destination

    size\_t destLength = 0;

    while (destination[destLength] != '\0') {

        ++destLength;

    }

    // Append the source to the destination

    for (size\_t i = 0; i < sourceLength; ++i) {

        destination[destLength++] = source[i];

    }

}

int main() {

    uint8\_t concatenatedMessage[MAX\_MESSAGE\_SIZE];

    size\_t totalLength = 0;

    int numFrames;

    // Read the number of frames

    printf("Enter the number of frames: ");

    scanf("%d", &numFrames);

    // Iterate through each frame

    for (int frame = 0; frame < numFrames; ++frame) {

        int numDataBits;

        // Read the size of the frame

        printf("Enter the size of frame %d: ", frame + 1);

        scanf("%d", &numDataBits);

        // Validate frame size

        if (numDataBits <= 1 || numDataBits > 8) {

            printf("Invalid size for frame %d. It should be between 2 and 8 bits.\n", frame + 1);

            return 1;

        }

        // Add frame size as the first element

        concatenatedMessage[totalLength++] = numDataBits;

        // Read data bits for the frame

        printf("Enter %d data bits for frame %d: ", numDataBits - 1, frame + 1);

        for (int i = 0; i < numDataBits - 1; ++i) {

            scanf("%hhu", &concatenatedMessage[totalLength]);

            ++totalLength;

        }

    }

    // Add NULL character to terminate the concatenated message

    concatenatedMessage[totalLength] = '\0';

    // Example of printing the concatenated message

    printf("Final concatenated data: ");

    for (size\_t i = 0; i < totalLength; ++i) {

        printf("%hhu ", concatenatedMessage[i]);

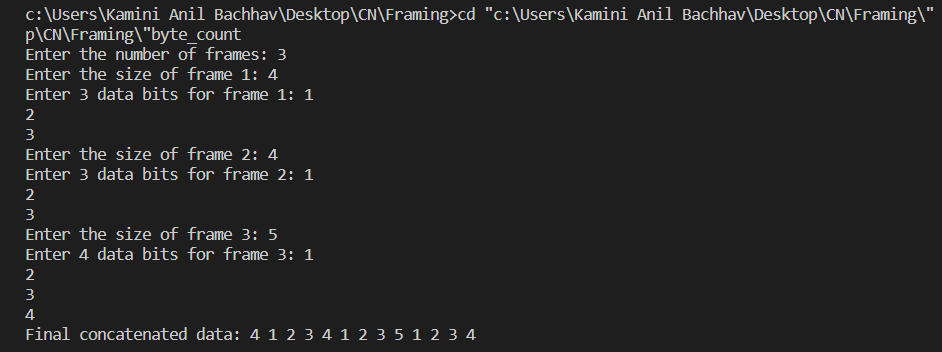
    }

    printf("\n");

    return 0;

}

**Output:**

****

**Code**: **FLAG BYTE WITH BYTE STUFFING**

#include<stdio.h>

#include<string.h>

main(){

        char a[20],b[20];

        int i,n,j;

        char f,s;

        printf("Enter the size of the frame : ");

        scanf("%d",&n);

        n=n\*2;

        printf("\nEnter the characters in frame : \n");

        for(i=0;i<n;i++)

                scanf("%c",&a[i]);

        printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\nFRAME:");

        for(i=0;i<n;i++)

                printf("%c",a[i]);

        j=0;

        for(i=0;i<n;i++)

        {

                if(a[i]=='f')

                {

                        b[j]='s';

                        j++;

                        b[j]=a[i];

                }

                else if(a[i]=='s')

                {

                        b[j]='s';

                        j++;

                        b[j]=a[i];

                }

                else

                        b[j]=a[i];

                j++;

        }

        printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\nRESULT:\n");

        printf("f");

        for(i=0;i<j;i++)

        {

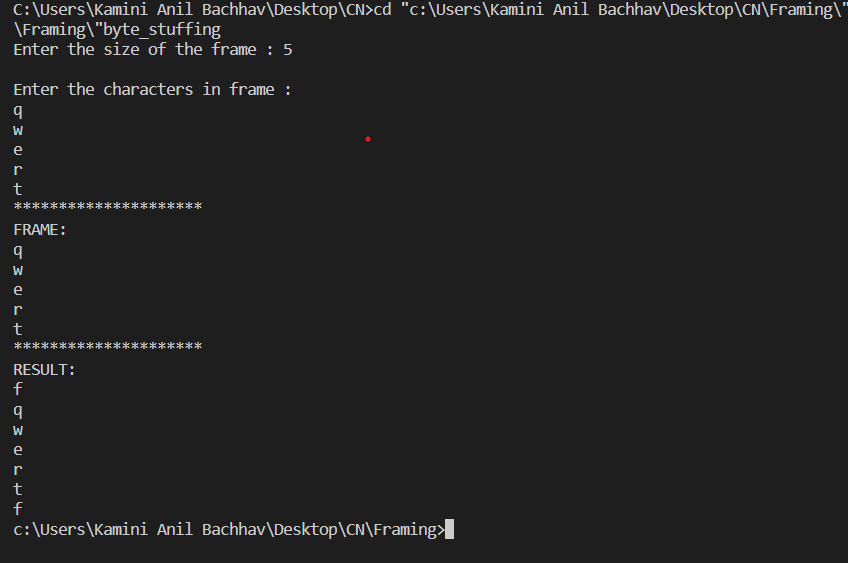
                printf("%c",b[i]);

        }

        printf("\nf");

}

**Output:**



**Code**: **FLAG BYTE WITH BIT STUFFING**

#include<stdio.h>

void main()

{

        int data[20];

        int stuff[30];

        int i,a=0,b,j,n;

        int count=0;

        printf("Enter the Numbers of bits to be send: ");

        scanf("%d",&n);

        printf("Enter the data \n");

        for(i=0;i<n;i++)

        {

                scanf("%d",&data[i]);

        }

        printf("The stuffed bits are\n");

        printf("01111110 ");

        for(i=0,j=0;i<n;i++,j++)

        {

                stuff[j]=data[i];

                if(data[i]=1)

                {

                        count++;

                }

                else

                {

                        count=0;

                }

        }

        for(i=0;i<j;i++)

        {

                printf("%d",stuff[i]);

        }

        printf(" ");

        printf("01111110\n");

}

**Output:**

