# Experiment No. 5 Date:

Framing Methods

**Aim:-** To study different Framing methods.

# Theory:

1. **Bit Stuffing**:
   * Bit stuffing involves adding an extra '0' bit after every five consecutive '1's in the data to avoid misinterpretation.
   * Flags are added at the beginning and end of the data without actual stuffing (i.e., without inserting extra bits).
   * Bit stuffing helps in maintaining clock synchronization between sender and receiver by avoiding long sequences of identical bits.

# Example:

* + - Original Data: **01111110 11111111**
    - Bit-Stuffed Data: **01111110 0111111010 011111110 011111110**
    - Explanation: Flags '01111110' are added at the beginning and end of the data without any additional stuffing. Extra '0' bits are inserted after every five consecutive '1's to avoid misinterpretation.

# Byte Stuffing:

* + Byte stuffing uses an escape character to differentiate control characters from actual data bytes.
  + Byte stuffing is commonly used in protocols like PPP (Point-to-Point Protocol) and HDLC (High-Level Data Link Control) to frame data for transmission.
  + Byte stuffing ensures data integrity by avoiding conflicts with control characters such as flags or escape characters.

# Example:

* + - Original Data: **FLAG DATA FLAG ESC**
    - Byte-Stuffed Data: **FLAG ESC FLAG DATA ESC FLAG ESC ESC FLAG**
    - Explanation: Flags 'FLAG' are added at the start and end of the data. If the data contains a flag or an escape character ('ESC'), an escape character is added before it to distinguish it from the flag sequence.

Code:

1. Byte Stuffing #include<iostream> #include<string.h>

using namespace std;

int main()

{

string str,temp;

cout<<"Enter the string"<<endl; cin>>str;

int n=str.length();

string flag="01111110",esc="11100000"; str.insert(0,flag);

str.append("01111110");

cout<<"The Byte Stuffed String : ";

if(n<8)

{

cout<<str;

}

else{

for(int i=8 ;i<=n+8;i=i+8)

{

temp=str.substr(i,8);

if(flag==temp || temp==esc)

{

str.insert(i,esc); i=i+8;

}

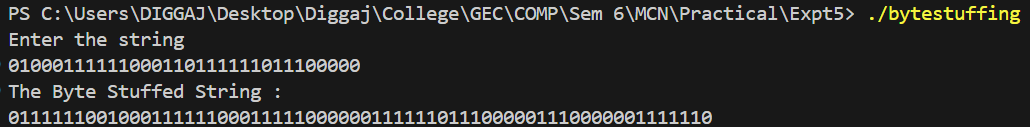
}

cout<<endl<<str;

}

}

Output



1. Bit Stuffing

#include<iostream>

#include<string.h>

using namespace std;

int main()

{

string str,temp;

cout<<"Enter the string"<<endl;

cin>>str;

int n=str.length(),cnt=0,l=0;

string flag="01111110";

str.insert(0,flag);

str.append("01111110");

for(int i=8;i<n+8+l;i++)

{

if(str[i]=='1')

{

cnt++;

if(cnt==6)

{

l++;

str.insert(i,"0"); cnt=0;

}

}

else{

cnt=0;

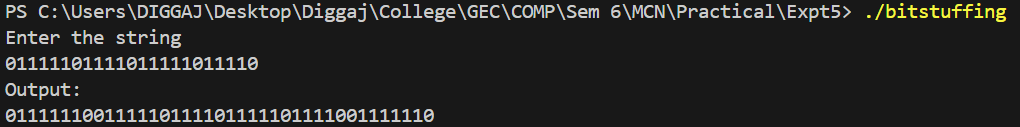
}

}

cout<<"Output:"<<endl<<str;

}

Output



Conclusion: Studied different Framing methods with successful execution of programs.