PROGRAM 1

Aim:- Write a C++ program to implement data link layer - Bit Stuffing.

Description:- Bit stuffing is the mechanism of inserting one or more non-information bits into a message to be transmitted, to break up the message sequence, for synchronization purpose. It is widely used in network and communication protocols, in which bit stuffing is a required part of the transmission process.

Bit Stuffing Mechanism

In a data link frame, the delimiting flag sequence generally contains six or more consecutive 1s. In order to differentiate the message from the flag in case of the same sequence, a single bit is stuffed in the message. Whenever a 0 bit is followed by five consecutive 1bits in the message, an extra 0 bit is stuffed at the end of the five 1s.

When the receiver receives the message, it removes the stuffed 0s after each sequence of five 1s. The un-stuffed message is then sent to the upper layers.

Algorithm:-

Step 1: Start

Step 2: Declare an array with max size of 100; variables n and new_n.

Step 3: Read length of input string. Check whether "n" (i.e. length of array) is greater than 0 or not. If it is less than 0, then print "Invalid size specified".

Step 4: Read n inputs from the user (i.e. message string) and store it into array.

Step 5: Traverse the array. Check for six consecutive ones and if they occur, stuff (insert) "0" after fifth one and increase the size of array (i.e., new_n++).And shift rest of the element to its next position (means after stuffed 0^{th} position) and perform this same step till whole array is traversed.

Step 6: At last, traverse the array upto new size of array (i.e., new_n) and then print the content of array and that will be our output after bit stuffing.

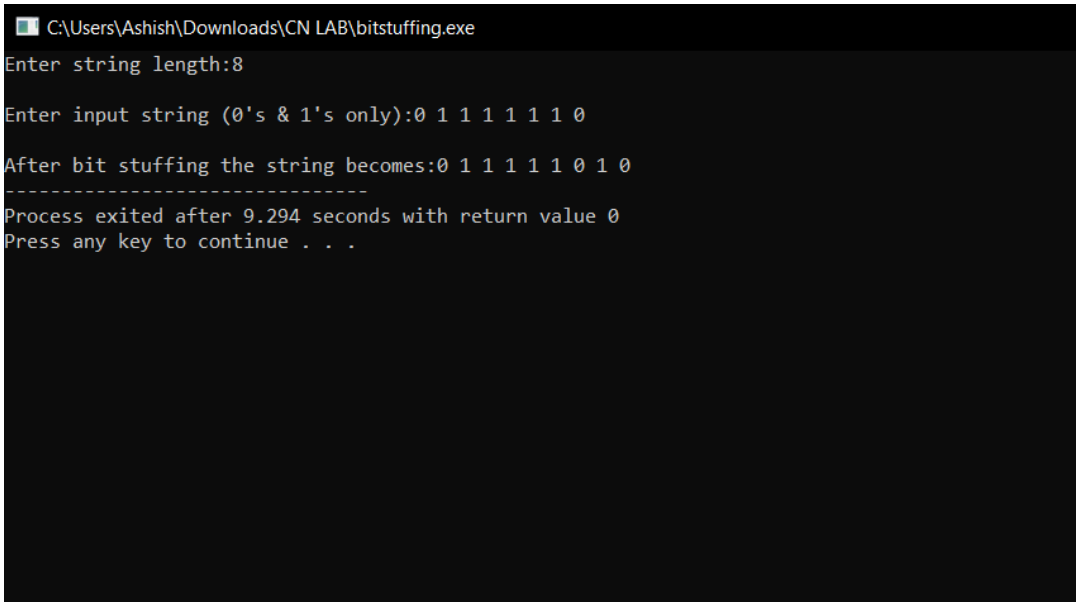
Step 7: Stop

CODE:-

```
#include<iostream>
#include<bits/stdc++.h>
#define MAX 100
using namespace std;

int main(){
    int i,j,a[MAX],n,new_n;
    cout<<"Enter string length:";
    cin>>n;
                if(n<=0){
                cout<<"Invalid size specified.\n";
                exit(0);
                }
                cout<<"\nEnter input string (0's & 1's only):";
                for(int i=0;i<n;i++)
                {
                cin>>a[i];   //storing each 0 & 1 in array
                }

    //Stuffing
    new_n=n;
    for(i=0;i<=MAX;i++){
        if(a[i]==0 && i+7 < new_n){
            if(a[i+1] == a[i+2] == a[i+3] == a[i+4] == a[i+5] == a[i+6] == 1 && a[i+7]==0) {
```

```
                new_n++;
                    for(j=new_n-1;j>i+6;j--){
                    a[j] = a[j-1];
                }
               a[i+6] = 0;    // stuff a bit 0
                i=i+8;
            }
        }
       }

   cout<<"\nAfter bit stuffing the string becomes:";
    for(i=0;i<new_n;i++){
     cout <<  a[i] << " ";
}
return 0;
}
```

OUTPUT:-



Discussion:- We have discussed what is data link layer, bit stuffing and its application while implementing this experiment and this is what we have found about it:

1. Data Link Layer is second layer of OSI Layered Model. This layer is one of the most complicated layers and has complex functionalities and liabilities. Data link layer is responsible for something called Framing, which is the division of stream of bits from network layer into manageable units (called frames). Frames could be of fixed size or variable size.
2. **Bit Stuffing** − A pattern of bits of arbitrary length is stuffed in the message to differentiate from the delimiter. This is also called bit - oriented framing.
3. **Applications of Bit Stuffing** – It helps to
   - synchronize several channels before multiplexing
   - rate-match two single channels to each other
   - run length limited coding

Learning & Finding:- We have successfully implemented bit stuffing and we learnt its application in computer network.