BIT Stuffing-

1. In the data layer, streams of bits from the physical layer are divided into data frames.
2. These data frames can be of fixed size or variable size.
3. In this process certain bits are added so that no pattern appears in the bits.
4. Whenever a pattern appears certain bits are added into the original bit.
5. It prevents the receiver from mistaking the data for control information.
6. It helps to ensure we are having accurate transmission of bits.

Process-

1. When the user enters the bits he or she wants to transmit.
2. First we travers through the bits.
3. Then user enter the pattern they want to be removed and give the value with which they want to remove.
4. Then we check the if pattern and bits match making the pattern.
5. If yes then the extra bit which we wanted to add gets added.
6. In this way we get the stuffed data.
7. The number of times this process will take place will form the frames.

CODE-

```
#include <iostream>
using namespace std;

string bitStuffing(const string &data, const string &pattern, char
stuffBit) {
    string stuffed = "";
    int count = 0;

    for (size_t i = 0; i < data.length(); i++) {
        stuffed += data[i];

        if (data[i] == pattern[count]) {
            count++;
        } else {
            count = (data[i] == pattern[0]) ? 1 : 0;
        }

        if (count == pattern.length()) {
            stuffed += stuffBit;
            count = 0;
        }
```

```
    }

    return stuffed;
}

string bitDeStuffing(const string &stuffed, const string &pattern) {
    string original = "";
    int count = 0;

    for (size_t i = 0; i < stuffed.length(); i++) {
        original += stuffed[i];

        if (stuffed[i] == pattern[count]) {
            count++;
        } else {
            count = (stuffed[i] == pattern[0]) ? 1 : 0;
        }

        if (count == pattern.length()) {
            i++;
            count = 0;
        }
    }

    return original;
}

int main() {
    int size, num;

    cout << "Enter the number of bits you want to transmit: ";
    cin >> num;

    cout << "Enter the size of the frame: ";
    cin >> size;

    int frameR = (num + size - 1) / size;
    cout << "Number of frames required: " << frameR << endl;

    string data, pattern;
```

```
    cout << "Enter the data bits: ";
    cin >> data;

    cout << "Enter the pattern to be stuffed: ";
    cin >> pattern;

    char stuffBit;
    cout << "Enter the bit to be inserted for stuffing (0 or 1): ";
    cin >> stuffBit;

    string stuffedData = bitStuffing(data, pattern, stuffBit);
    cout << "\nStuffed Data: " << stuffedData << endl;

    int totalFrames = (stuffedData.length() + size - 1) / size;
    cout << "Total frames required (after stuffing): " << totalFrames <<
endl;

    string destuffedData = bitDeStuffing(stuffedData, pattern);
    cout << "De-stuffed Data: " << destuffedData << endl;

    return 0;
}
```

OUTPUT-

```
PS C:\upes\sem4\DCN_LAB> g++ stuffing.cpp -o main
PS C:\upes\sem4\DCN_LAB> ./main.exe
Enter the number of bits you want to transmit: 16
Enter the size of the frame: 4
Number of frames required: 4
Enter the data bits: 1010101000111101
Enter the pattern to be stuffed: 000
Enter the bit to be inserted for stuffing (0 or 1): 0

Stuffed Data: 10101010000111101
Total frames required (after stuffing): 5
De-stuffed Data: 1010101000111101
```