Ex No: 3 Implement the data link layer framing methods such as character, character stuffing and bit stuffing.

AIM:

To implement the data link layer framing methods such as character, character stuffing and bit stuffing.

ALGORITHM:

CHARACTER STUFFING

Step 1: Start the program.

Step 2: Add string to the program.

Step 3: Add character stuffing.

Step 4: Convert string to binary.

Step 5: Add Bit destuffing and Bit stuffing.

Step 6: Again, convert string to binary.

Step 7: End the program.

BIT STUFFING

Step 1: Start the program.

Step 2: Add data types to the program.

Step 3: Add character to get the output.

Step 4: Convert string and add string to the program.

Step 5: Add Bit destuffing.

Step 6: End the program.

Bit stuffing and destuffing

```
import java.util.*;
public class bit_stuffing
    {
    public static void main(String[] args)
     {
         Scanner sc=new Scanner(System.in);
         System.out.print("Enter the message:-");
         String d1 = sc.nextLine();
         String remaining = new String();
```

```
 String output=new String();
 int counter = 0;
 for(int i=0;i<d1.length();i++)
  {

    if (d1.charAt(i)!='1' && d1.charAt(i)!='0')
       {
          System.out.println("Enter valid Binary values");
          return;
       }
    if(d1.charAt(i) == '1')
       {
          counter++;
          remaining = remaining + d1.charAt(i);
       }
    else
       {
          remaining = remaining + d1.charAt(i);
          counter = 0;
       }
    if(counter == 5)
       {
          remaining = remaining + '0';
          counter = 0;
       }
  }
System.out.println("Flag--> 01111110");
String new1="|01111110 | "+remaining+" | 01111110|";
System.out.println("Stuffed data at intermediate site is:");
for(int k=0;k<=(28+d1.length());k++)
{
 System.out.print("-");
}
System.out.println();
System.out.println(" "+new1);
for(int k=0;k<=(28+d1.length());k++)
{
 System.out.print("-");
}
System.out.println();
counter=0;
for(int i=0;i<remaining.length();i++)
  {

     if(remaining.charAt(i) == '1')
       {
```

```
                counter++;
                output = output + remaining.charAt(i);

            }
          else
            {
                output = output + remaining.charAt(i);
                counter = 0;
            }
          if(counter == 5)
            {
                if((i+2)!=remaining.length())
                {
                  output = output + remaining.charAt(i+2);
                }
                else
                {
                  output=output + '1';
                }
                i=i+2;
                counter = 1;
            }
        }
       System.out.println("Destuffed BIT is: "+output);
    }
}
```

SAMPLE OUTPUT

```
Enter the message:-011111110110
Flag--> 01111110
Stuffed data at intermediate site is:
----------------------------------------
|01111110 | 0111110110110 | 01111110|
----------------------------------------
Destuffed BIT is: 011111110110
```

CHARACTER STUFFING AND DESTUFFING

```
import java.util.*;

public class CharacterStuffingDemo {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);

      System.out.print("Enter the data to be stuffed: ");
      String input = scanner.nextLine();
```

```
    String stuffedData = characterStuffing(input);
    System.out.println("Stuffed data: " + stuffedData);

    String destuffedData = characterDestuffing(stuffedData);
    System.out.println("Destuffed data: " + destuffedData);

    scanner.close();
}

public static String characterStuffing(String input) {
    char flag = '/';
    char escape = 'E';

    StringBuilder stuffedData = new StringBuilder();
    stuffedData.append(flag);

    for (char c : input.toCharArray()) {
        if (c == '/' || c == 'E') {
            stuffedData.append(escape);
        }
        stuffedData.append(c);
    }

    stuffedData.append(flag);
    return stuffedData.toString();
}

public static String characterDestuffing(String stuffedData) {
    char flag = '/';
    char escape = 'E';

    StringBuilder destuffedData = new StringBuilder();

    char[] characters = stuffedData.toCharArray();
    boolean escapeMode = false;

    for (int i = 1; i < characters.length - 1; i++) {
        char c = characters[i];

        if (escapeMode) {
            destuffedData.append(c);
            escapeMode = false;
        } else if (c == escape) {
            escapeMode = true;
        } else {
            destuffedData.append(c);
```

```
            }
        }

        return destuffedData.toString();
    }
}
```

SAMPLE OUTPUT

```
Enter the data to be stuffed: AEFGHTJ
Stuffed data: /AEEFGHTJ/
Destuffed data: AEFGHTJ
```