	Subject :		Software :	
			Hardware :	
	Branch :	Semester :	Page No. 11	Prog No. 02

PROBLEM STATEMENT: Impliment the data link layer framing methods such as character, character stuffing and bit stuffing in C.

ALGORITHM & CODE:

1. Character framing:

In character framing, special characters (like a start and stop character) are used to mark the beginning and end of a frame.

2. Character Stuffing:

Character stuffing is used to ensure that the special characters used for framing don't appear within the data. If the special character appears in the data, an extra "escape" character is inserted before it.

3. Bit Stuffing:

Bit stuffing ensures that a sequence of consecutive bits (like 5 consecutive 1s) does not appear in the data stream. If such a sequence appears, a 0 bit is inserted after 5 consecutive 1s.

1. Character framing in C:

Here is an implementation of character framing using a START and STOP character.

INPUT GIVEN

OUTPUT OBTAINED

REMARKS

GRADE : | Signature of Faculty Date : | Signature of Student Date :

	Subject :		Software :	
			Hardware :	
	Branch :	Semester :	Page No. 12	Prog No.
PROBLEM STATEMENT				

ALGORITHM & CODE :

```
# include <stdio.h>
# include <string.h>

# define START_CHAR 'S'
#define STOP_CHAR 'E'
#define ESCAPE_CHAR 'X'

void apply_charactor_stuffing (char* data){
    Printf ("%c", START_CHAR);

for (int i = 0; i < strlen (data); i++){
    if (data [i] == START_CHAR || data [i] == STOP_CHAR){
       Printf ("%c", ESCAPE_CHAR); //Stuffing the escape charactor
    }
    Printf ("%c", data [i]);
  }
 Printf ("%c", STOP_CHAR);
}

int main (){
    Char data[] = "HelSloWorld";
    Printf ("Original data : %s\n", data);
    Printf ("Stuffed data:");
    apply_Charactor_Stuffing (data);
    return 0;
}
```

INPUT GIVEN		
OUTPUT OBTAINED		
REMARKS		
GRADE :	Signature of Faculty Date :	Signature of Student Date :

Subject :		Software :
		Hardware :
Branch :	Semester :	Page No. 13 Prog No.

PROBLEM STATEMENT

ALGORITHM & CODE: 3. Bit Stuffing in C:

In Bit Stuffing, we need to monitor the bit sequence and insert a 0 after five consecutive 1s.
Here's how we can do it.

```
#include <stdio.h>
void apply_bit-stuffing (char* data){
    int bit-count = 0;
    Printf ("Framed data with bit Stuffing: ");
    for (int i = 0; data[i] ! = '\0'; i++){
        unsigned char byte = data[i];
        // check each bit in the byte
        for (int j = 7; j>= 0; j- ){
            int bit = (byte >>j)& 1;
            Printf ("%d', bit);
      // if we encounter 5 consecutive 1s, insert a 0
    if (bit = = 1){
        bit_count ++;
        if (bit- count = = 5){
            Printf ("0"); // stuff a zero after 5 consecutive ones
            bit_count = 0; //Reset count
      }
    }else {
```

INPUT GIVEN

OUTPUT OBTAINED

REMARKS

GRADE: Signature of Faculty Date: Signature of Student Date:

Subject :		Software :	
		Hardware :	
Branch :	Semester :	Page No. 14	Prog No.

PROBLEM STATEMENT

ALGORITHM & CODE :

```
        bit_count = 0; // Reset count on 0
      }
    }
  }
}

int main(){
    char data[]= "HelloWorld";
    printf ("Original Data: %s \n", data);
    apply_bit_stuffing (data);
    return 0;
}
```

Explanation of Each method:

1. Character framing: We simplify wrap the data with the START_CHAR and STOP_CHAR.
   It is straight forward and used when special characters (such as S and E) are reserved for this purpose.

2. Character stuffing: We loop through the data and check for START_CHAR or STOP_CHAR. If we encounter these we stuff them with an ESCAPE_CHAR to ensure they don't interfere with the framing characters.

INPUT GIVEN

OUTPUT OBTAINED

REMARKS

GRADE :	Signature of Faculty	Signature of Student
	Date :	Date :

	Subject :		Software :	
			Hardware :	
	Branch :	Semester :	Page No. 15	Prog No.

PROBLEM STATEMENT

ALGORITHM & CODE : 3. Bit Stuffing: We read each byte of data and examine its bits. If five consecutive 1 bits are found. we insert a 0 to prevent this Pattern from being misinterpreted as the end of a frame.

Sample Output of each Program:

Character framing:

~~Kotlin~~
~~Copy~~

```
Original data: HelloWorld
Framed data: SHEllo WorldE
```

Character Stuffing:

```
Original Data: HelSlo WorldE
Stuffed Data: XSHello XWorld XE
```

Bit Stuffing:

```
Original Data: HelloWorld
Framed Data with Bit Stuffing:
01001 000011 00 1010 11011000110110001 10111100 1000000
1010 11 10110 1111011101000 11011 0001100 100
```

INPUT GIVEN

OUTPUT OBTAINED

REMARKS

GRADE : Signature of Faculty
Date :

Signature of Student
Date :