

Program:

```
//Bit Stuffing #include<stdio.h> #include<string.h> #define max 100 void main()
{
int n,j=0,i=0,count;

printf("Enter Number of Frames : "); scanf("%d",&n);

char frame[max][max]; //frame is a 2D array which holds n no. of bit sequence for(i=0;i<n;i++)
{
printf("\nEnter Frame %d:\n",i+1); scanf("%s",frame[i]);
}

PVP SIDDHARTHA INSTITUTE OF TECHNOLOGY

45

printf("The transmitted frame is :\n\n"); for(i=0;i<n;i++)
{
count=0;

printf("01111110 "); //At the start of each frame flag(01111110) is stuffed for(j=0 ; frame[i][j]!='\0' ; j++)
{
printf("%c",frame[i][j]);

if(frame[i][j]=='1') count++;

else

count=0;

if(count==5) //if there are five consecutive 1s in data --zero is stuffed
{
printf("0"); //At the end of each frame flag(01111110) is stuffed count=0;
}
}
}
```

```

printf(" 01111110");
}
}

//Character stuffing: #include<stdio.h> #include<string.h> void main()
{
int n,i,j,k,count;

char flag[10],tmp[10]; char escape[10];

char frame[10][20],temp[10][20]; printf("Enter Number of Frames: ");

scanf("%d",&n);

printf("Enter Flag:\n"); scanf("%s",&flag); while(1)
{
if(strlen(flag)==8) break;

else

{

printf("Re-Enter flag of 8 bits :"); scanf("%s",&flag);

}

}

printf("Enter Escape Character:\n");

PVP SIDDHARTHA INSTITUTE OF TECHNOLOGY

46

scanf("%s",escape);

while(1){ if(strlen(flag)==8) break;

else{

printf("Re-Enter escape of 8 bits :"); scanf("%s",&flag);

}

```

```

}
for(i=0;i<n;i++)
{
printf("Enter Frame %d : \n",i+1); scanf("%s",frame[i]); while(strlen(frame[i])%8!=0)
{
printf("Re-enter the Frame %d of length--Multiples of 8 : ",i+1); scanf("%s",frame[i]);
}
}
printf("The transmitted frame is:\n\n"); for(i=0;i<n;i++)
{
printf("%s",flag); for(j=0;j<(strlen(frame[i]))/8;j++) {
count=0; for(k=j*8;count<8;k++){
tmp[count]=frame[i][k]; count++;
}
tmp[count]='\0'; if(!strcmp(tmp,flag))
printf("%s",escape); else if(!strcmp(tmp,escape))
printf("%s",escape); printf("%s",tmp);
}
printf("%s",flag);
}
}

```

PVP SIDDHARTHA INSTITUTE OF TECHNOLOGY

47

Output (Test Cases): Character Stuffing:

Output (Test Cases):
Character Stuffing:

Input	Output	Executed / Not Executed
Enter Number of Frames: 1 Enter Flag: 10101010 Enter Escape Character: 11110000 Enter Frame 1: 0011001110101010	The transmitted frame is: 10101010001100111111000 0 1010101010101010	Executed
Enter Number of Frames: 1 Enter Flag: 10101010 Enter Escape Character: 11110000 Enter Frame 1: 1111000000110011	The transmitted frame is: 10101010111100001111000 0 0011001110101010	Executed
Enter Number of Frames: 2 Enter Flag: 10101010 Enter Escape Character: 11110000 Enter Frame 1: 0011001110101010 Enter Frame 2: 1111000010101010	The transmitted frame is: 10101010001100111111000 0 10101010101010101010101 0 11110000111100001111000 0 1010101010101010	Executed

Bit Stuffing:

Input	Output	Executed / Not Executed
Enter Number of Frames: 1 Enter Frame 1: 001100111110101010	The transmitted frame is: 01111110 00110011111001010100111110	Executed
Enter Number of Frames: 2 Enter Frame 1: 001100111110101010 Enter Frame 2: 1111111111	The transmitted frame is: 01111110 0011001111100101010 0111111001111110111110111110 01111110	Executed

Result:

Thus the program for Framing i.e. bit stuffing and character stuffing is executed