Name : Sahil Dattatray Mohite

Div : SY-IT-A Batch : B1

Roll No. 30 PRN : 12010501

Computer Network – Lab Assignment 3

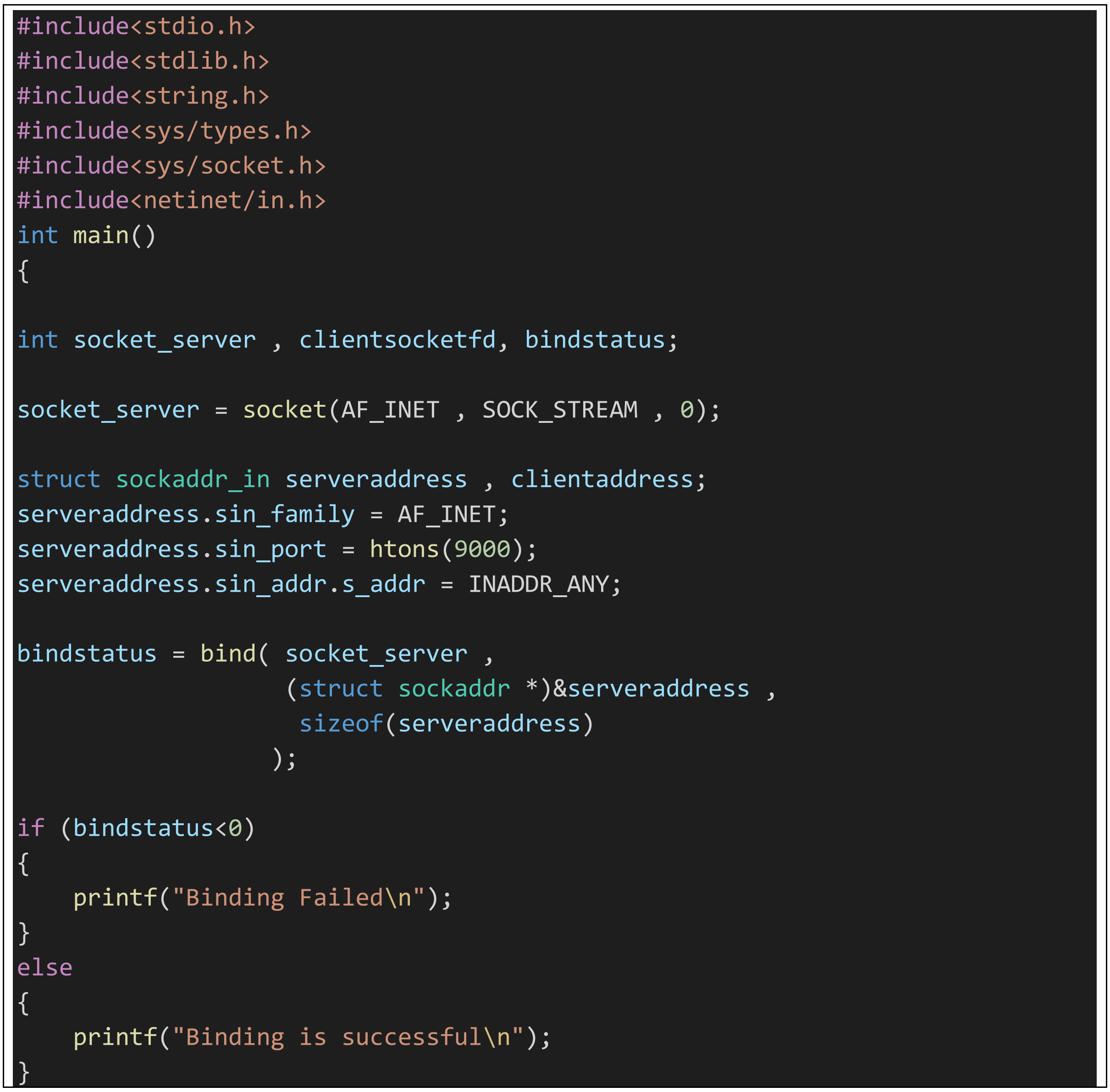
Unit : Framing Methods

Question:

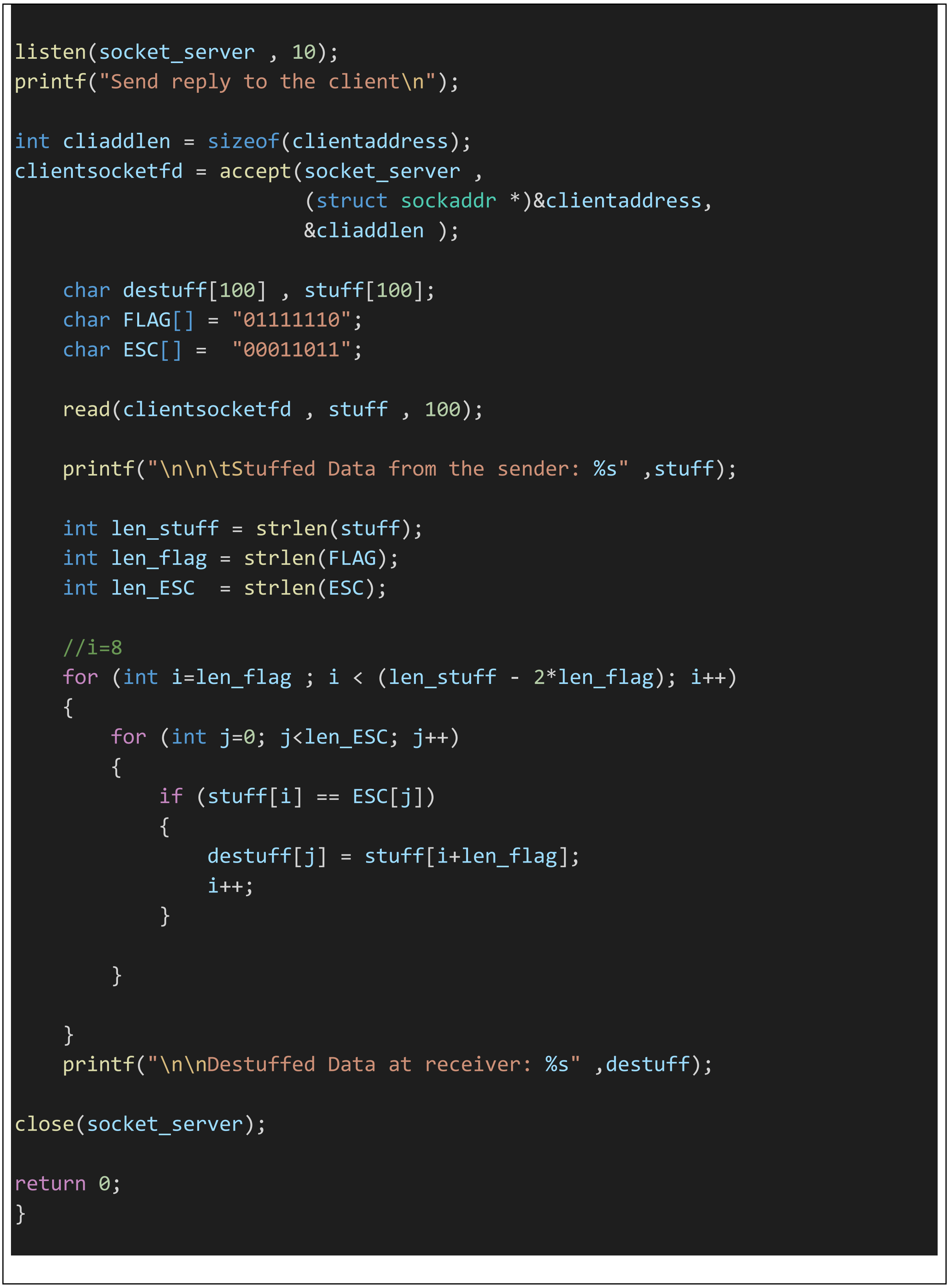
Implement FLAG Byte with Byte Stuffing and FLAG Byte with Bit stuffing framing method using Client -server communication.(TCP/UDP Socket)

A)FLAG Byte With Byte Stuffing

Byte_server.c

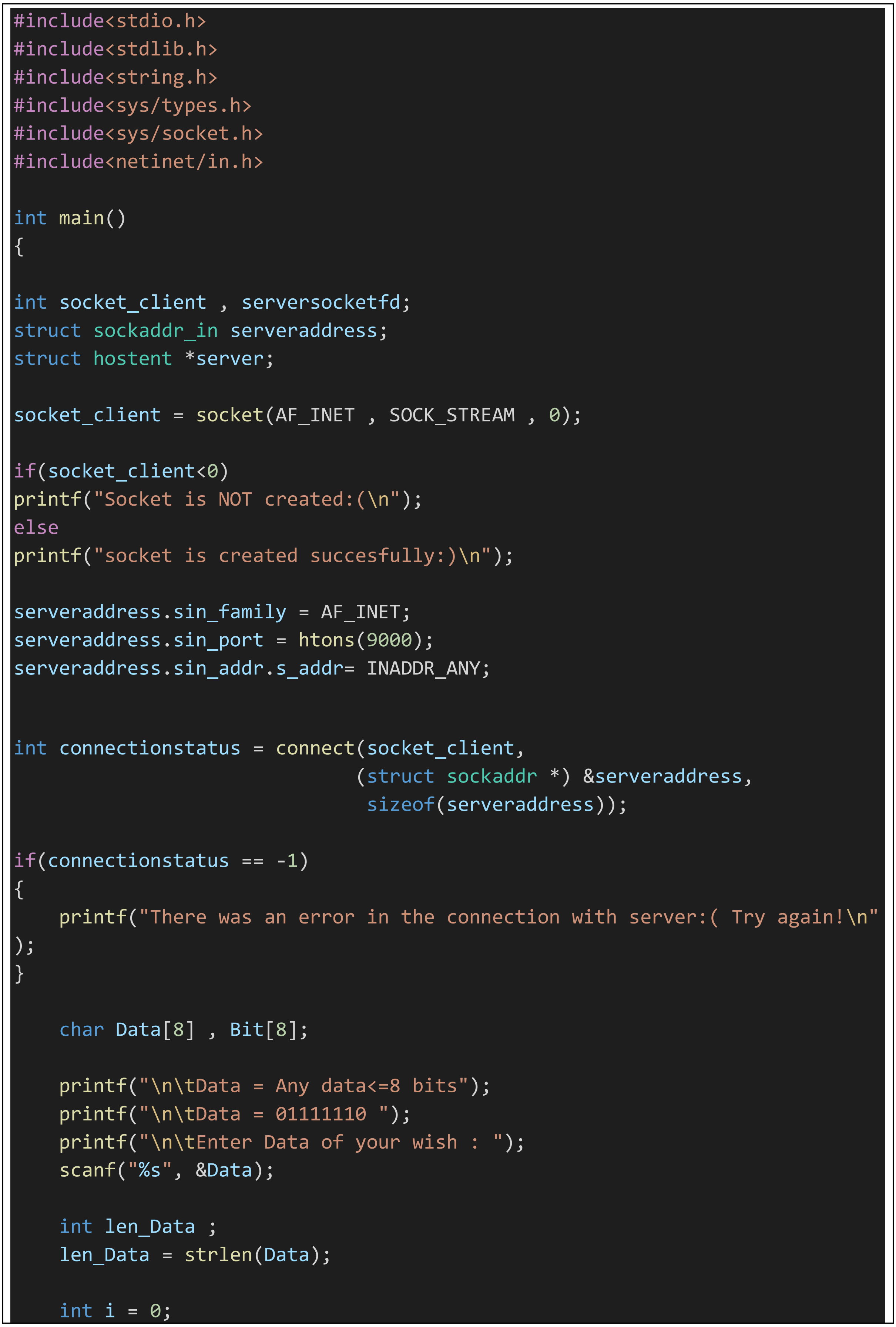


```c
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<netinet/in.h>
int main()
{

int socket_server , clientsocketfd, bindstatus;

socket_server = socket(AF_INET , SOCK_STREAM , 0);

struct sockaddr_in serveraddress , clientaddress;
serveraddress.sin_family = AF_INET;
serveraddress.sin_port = htons(9000);
serveraddress.sin_addr.s_addr = INADDR_ANY;

bindstatus = bind( socket_server ,
                   (struct sockaddr *)&serveraddress ,
                    sizeof(serveraddress)
                  );

if (bindstatus<0)
{
    printf("Binding Failed\n");
}
else
{
    printf("Binding is successful\n");
}
```

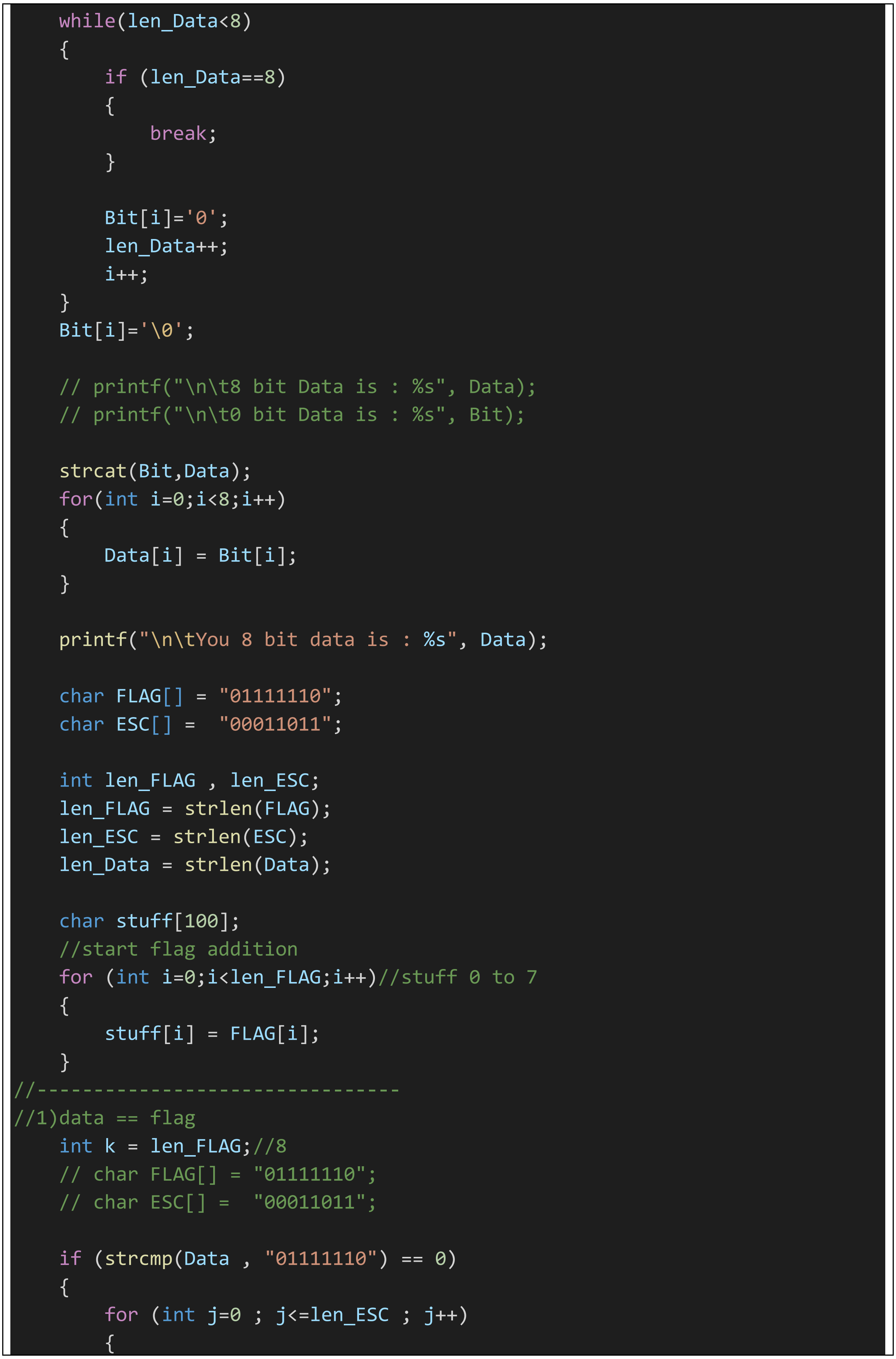


```
listen(socket_server , 10);
printf("Send reply to the client\n");

int cliaddlen = sizeof(clientaddress);
clientsocketfd = accept(socket_server ,
                        (struct sockaddr *)&clientaddress,
                        &cliaddlen );

    char destuff[100] , stuff[100];
    char FLAG[] = "01111110";
    char ESC[] =  "00011011";

    read(clientsocketfd , stuff , 100);

    printf("\n\n\tStuffed Data from the sender: %s" ,stuff);

    int len_stuff = strlen(stuff);
    int len_flag = strlen(FLAG);
    int len_ESC  = strlen(ESC);

    //i=8
    for (int i=len_flag ; i < (len_stuff - 2*len_flag); i++)
    {
        for (int j=0; j<len_ESC; j++)
        {
            if (stuff[i] == ESC[j])
            {
                destuff[j] = stuff[i+len_flag];
                i++;
            }

        }

    }
    printf("\n\nDestuffed Data at receiver: %s" ,destuff);

close(socket_server);

return 0;
}
```

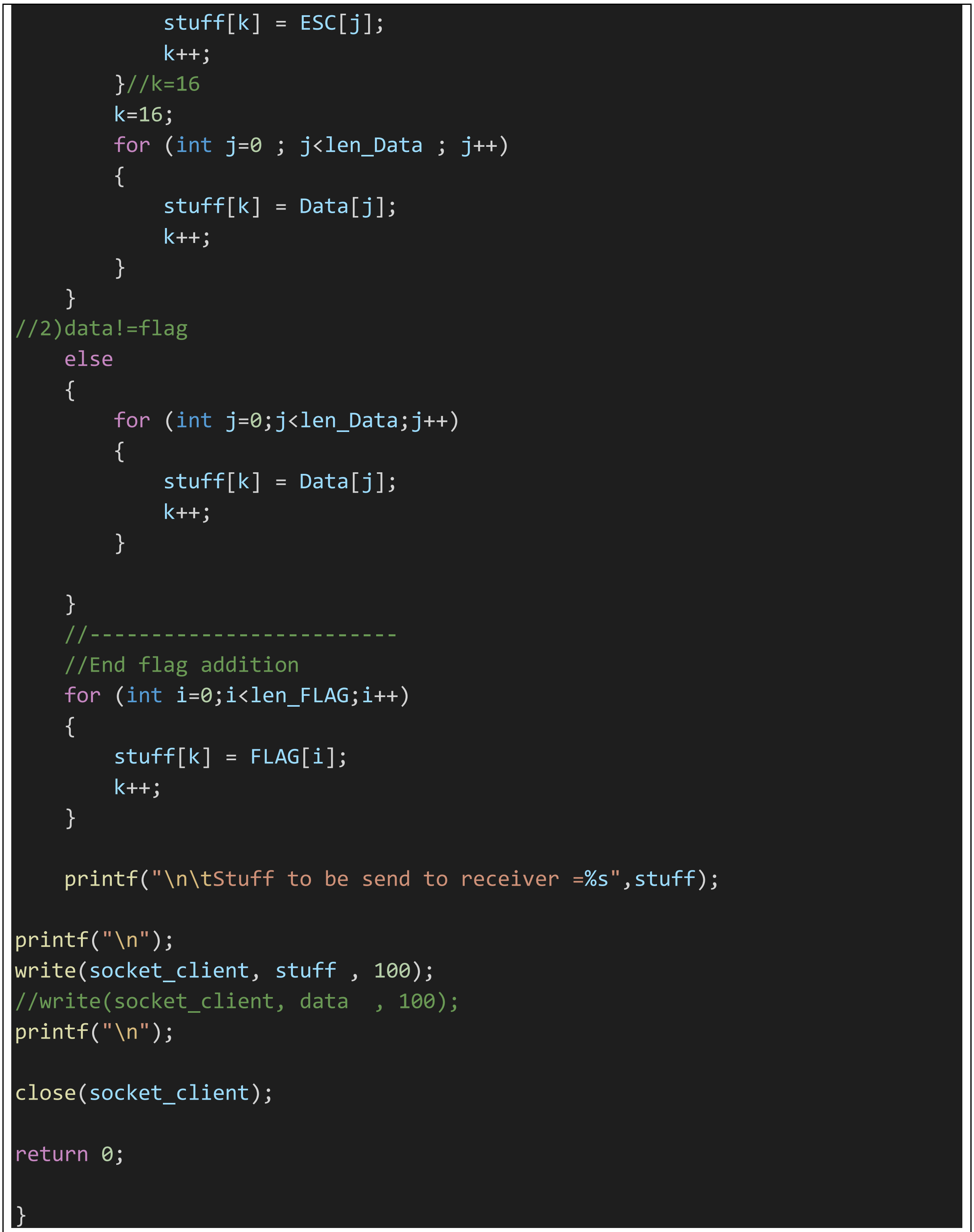
Byte_client.c



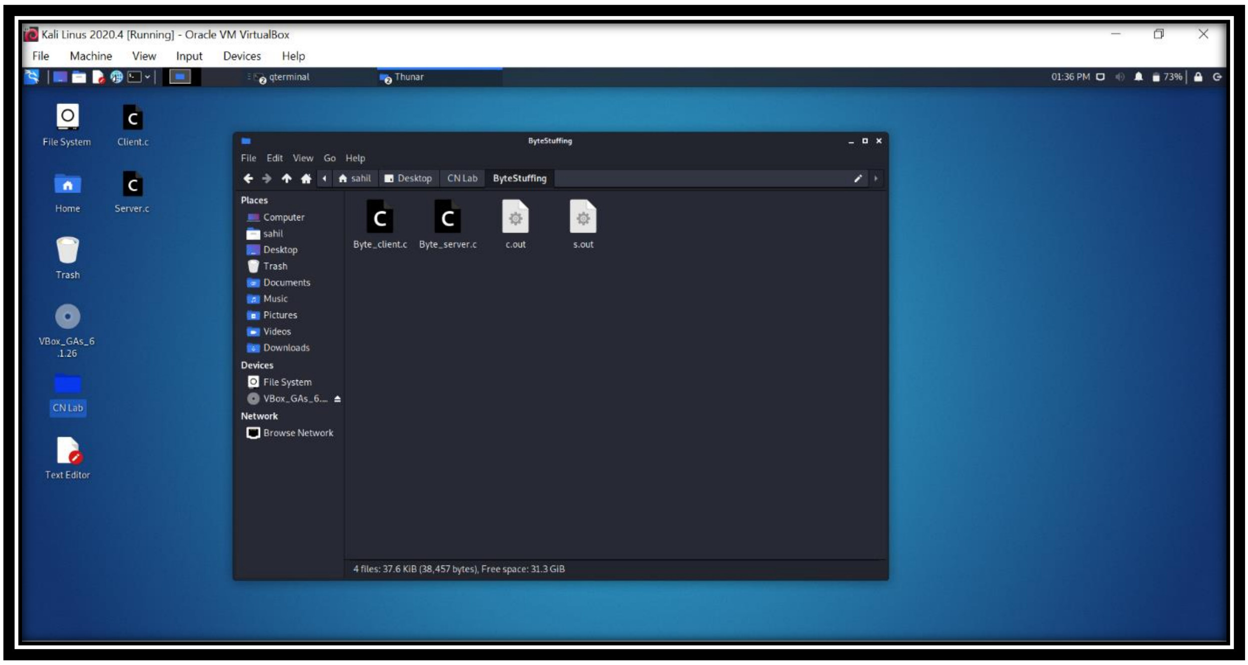
```c
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<netinet/in.h>

int main()
{

int socket_client , serversocketfd;
struct sockaddr_in serveraddress;
struct hostent *server;

socket_client = socket(AF_INET , SOCK_STREAM , 0);

if(socket_client<0)
printf("Socket is NOT created:(\n");
else
printf("socket is created succesfully:)\n");

serveraddress.sin_family = AF_INET;
serveraddress.sin_port = htons(9000);
serveraddress.sin_addr.s_addr= INADDR_ANY;


int connectionstatus = connect(socket_client,
                              (struct sockaddr *) &serveraddress,
                               sizeof(serveraddress));

if(connectionstatus == -1)
{
    printf("There was an error in the connection with server:( Try again!\n"
);
}

    char Data[8] , Bit[8];

    printf("\n\tData = Any data<=8 bits");
    printf("\n\tData = 01111110 ");
    printf("\n\tEnter Data of your wish : ");
    scanf("%s", &Data);

    int len_Data ;
    len_Data = strlen(Data);

    int i = 0;
```



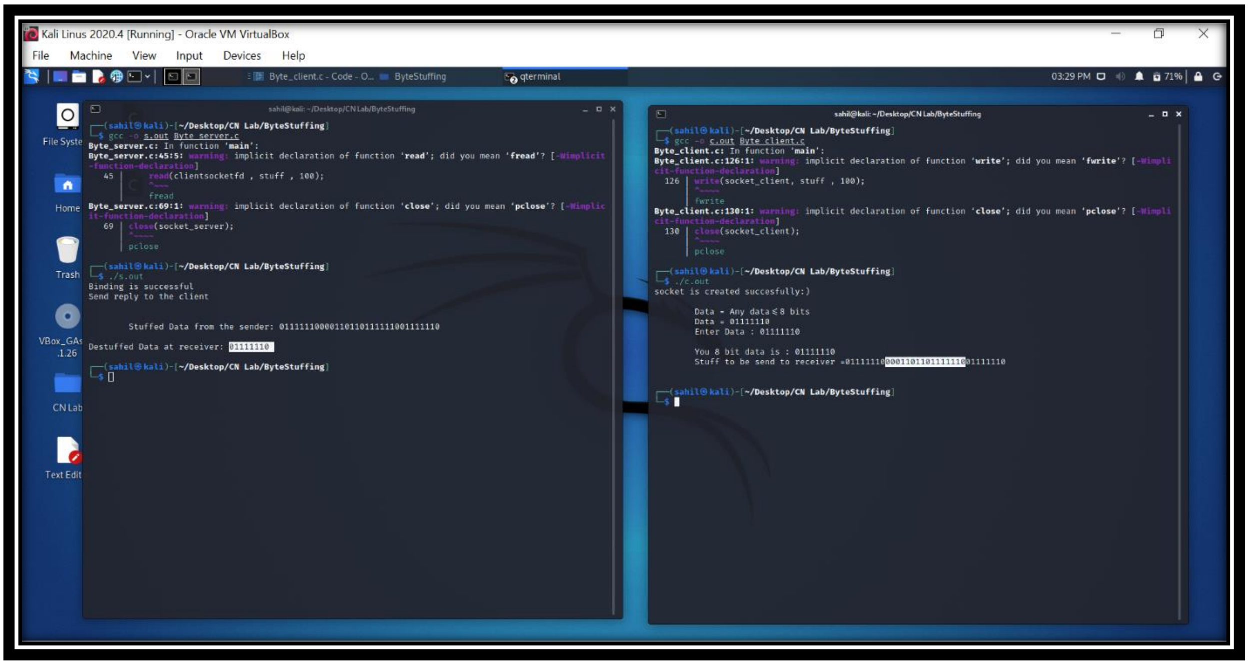
```
    while(len_Data<8)
    {
        if (len_Data==8)
        {
            break;
        }

        Bit[i]='0';
        len_Data++;
        i++;
    }
    Bit[i]='\0';

    // printf("\n\t8 bit Data is : %s", Data);
    // printf("\n\t0 bit Data is : %s", Bit);

    strcat(Bit,Data);
    for(int i=0;i<8;i++)
    {
        Data[i] = Bit[i];
    }

    printf("\n\tYou 8 bit data is : %s", Data);

    char FLAG[] = "01111110";
    char ESC[] =  "00011011";

    int len_FLAG , len_ESC;
    len_FLAG = strlen(FLAG);
    len_ESC = strlen(ESC);
    len_Data = strlen(Data);

    char stuff[100];
    //start flag addition
    for (int i=0;i<len_FLAG;i++)//stuff 0 to 7
    {
        stuff[i] = FLAG[i];
    }
//--------------------------------
//1)data == flag
    int k = len_FLAG;//8
    // char FLAG[] = "01111110";
    // char ESC[] =  "00011011";

    if (strcmp(Data , "01111110") == 0)
    {
        for (int j=0 ; j<=len_ESC ; j++)
        {
```



```
            stuff[k] = ESC[j];
            k++;
        }//k=16
        k=16;
        for (int j=0 ; j<len_Data ; j++)
        {
            stuff[k] = Data[j];
            k++;
        }
    }
//2)data!=flag
    else
    {
        for (int j=0;j<len_Data;j++)
        {
            stuff[k] = Data[j];
            k++;
        }

    }
    //-------------------------
    //End flag addition
    for (int i=0;i<len_FLAG;i++)
    {
        stuff[k] = FLAG[i];
        k++;
    }

    printf("\n\tStuff to be send to receiver =%s",stuff);

printf("\n");
write(socket_client, stuff , 100);
//write(socket_client, data  , 100);
printf("\n");

close(socket_client);

return 0;

}
```



4 files: 37.6 KiB (38,457 bytes), Free space: 31.3 GiB


Kali Linus 2020.4 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Byte_client.c - Code - O...
ByteStuffing
qterminal
03:29 PM
71%
File System
Home
Trash
VBox_GAs_6.1.26
CN Lab
Text Editor
sahil@kali: ~/Desktop/CN Lab/ByteStuffing

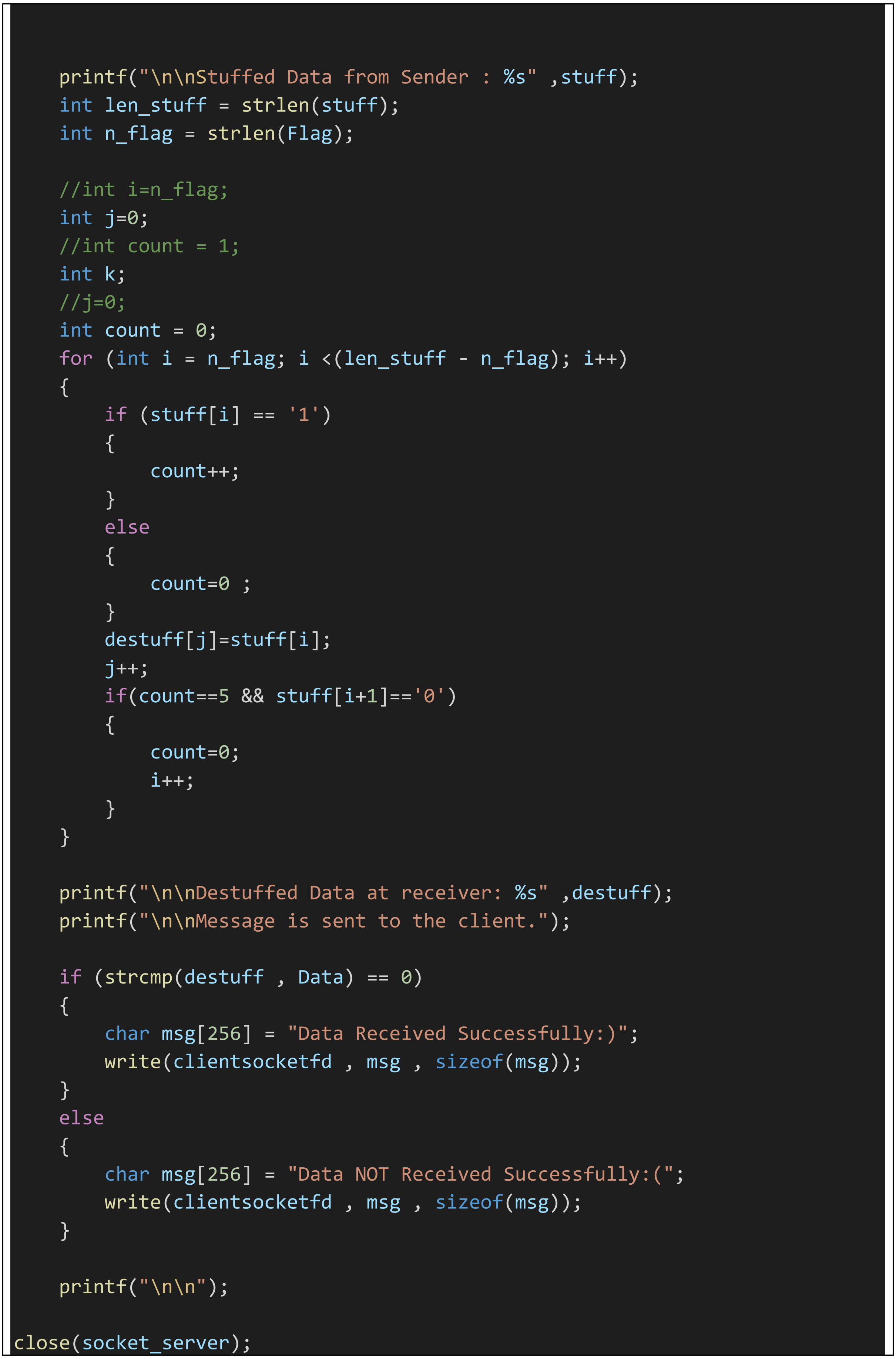
```
┌──(sahil㉿kali)-[~/Desktop/CN Lab/ByteStuffing]
└─$ gcc -o s.out Byte_server.c
Byte_server.c: In function 'main':
Byte_server.c:45:5: warning: implicit declaration of function 'read'; did you mean 'fread'? [-Wimplicit
-function-declaration]
   45 |     read(clientsocketfd , stuff , 100);
      |     ^~~~
      |     fread
Byte_server.c:69:1: warning: implicit declaration of function 'close'; did you mean 'pclose'? [-Wimplic
it-function-declaration]
   69 | close(socket_server);
      | ^~~~~
      | pclose

┌──(sahil㉿kali)-[~/Desktop/CN Lab/ByteStuffing]
└─$ ./s.out
Binding is successful
Send reply to the client


        Stuffed Data from the sender: 01111110000110110111111001111110

Destuffed Data at receiver: 01111110

┌──(sahil㉿kali)-[~/Desktop/CN Lab/ByteStuffing]
└─$ 
```

```
┌──(sahil㉿kali)-[~/Desktop/CN Lab/ByteStuffing]
└─$ gcc -o c.out Byte_client.c
Byte_client.c: In function 'main':
Byte_client.c:126:1: warning: implicit declaration of function 'write'; did you mean 'fwrite'? [-Wimpli
cit-function-declaration]
  126 | write(socket_client, stuff , 100);
      | ^~~~~
      | fwrite
Byte_client.c:130:1: warning: implicit declaration of function 'close'; did you mean 'pclose'? [-Wimpli
cit-function-declaration]
  130 | close(socket_client);
      | ^~~~~
      | pclose

┌──(sahil㉿kali)-[~/Desktop/CN Lab/ByteStuffing]
└─$ ./c.out
socket is created succesfully:)

        Data - Any data ≤ 8 bits
        Data = 01111110
        Enter Data : 01111110

        You 8 bit data is : 01111110
        Stuff to be send to receiver =01111110000110110111111001111110


┌──(sahil㉿kali)-[~/Desktop/CN Lab/ByteStuffing]
└─$ 
```

B)FLAG Byte With Bit Stuffing

Bit_server.c

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<netinet/in.h>
int main()
{

int socket_server , clientsocketfd, bindstatus;

socket_server = socket(AF_INET , SOCK_STREAM , 0);

struct sockaddr_in serveraddress , clientaddress;
serveraddress.sin_family = AF_INET;
serveraddress.sin_port = htons(9000);
serveraddress.sin_addr.s_addr = INADDR_ANY;

bindstatus = bind( socket_server ,
                   (struct sockaddr *)&serveraddress ,
                    sizeof(serveraddress)
                  );

if (bindstatus<0)
{
    printf("Binding Failed\n");
}
else
{
    printf("Binding is successful\n");
}

listen(socket_server , 10);
printf("Send reply to the client\n");

int cliaddlen = sizeof(clientaddress);
clientsocketfd = accept(socket_server ,
                        (struct sockaddr *)&clientaddress,
                        &cliaddlen );

    char destuff[100] , stuff[100] , Data[100];
    char Flag[] = "01111110";
    read(clientsocketfd , stuff , 100);
    read(clientsocketfd , Data, 100);
```

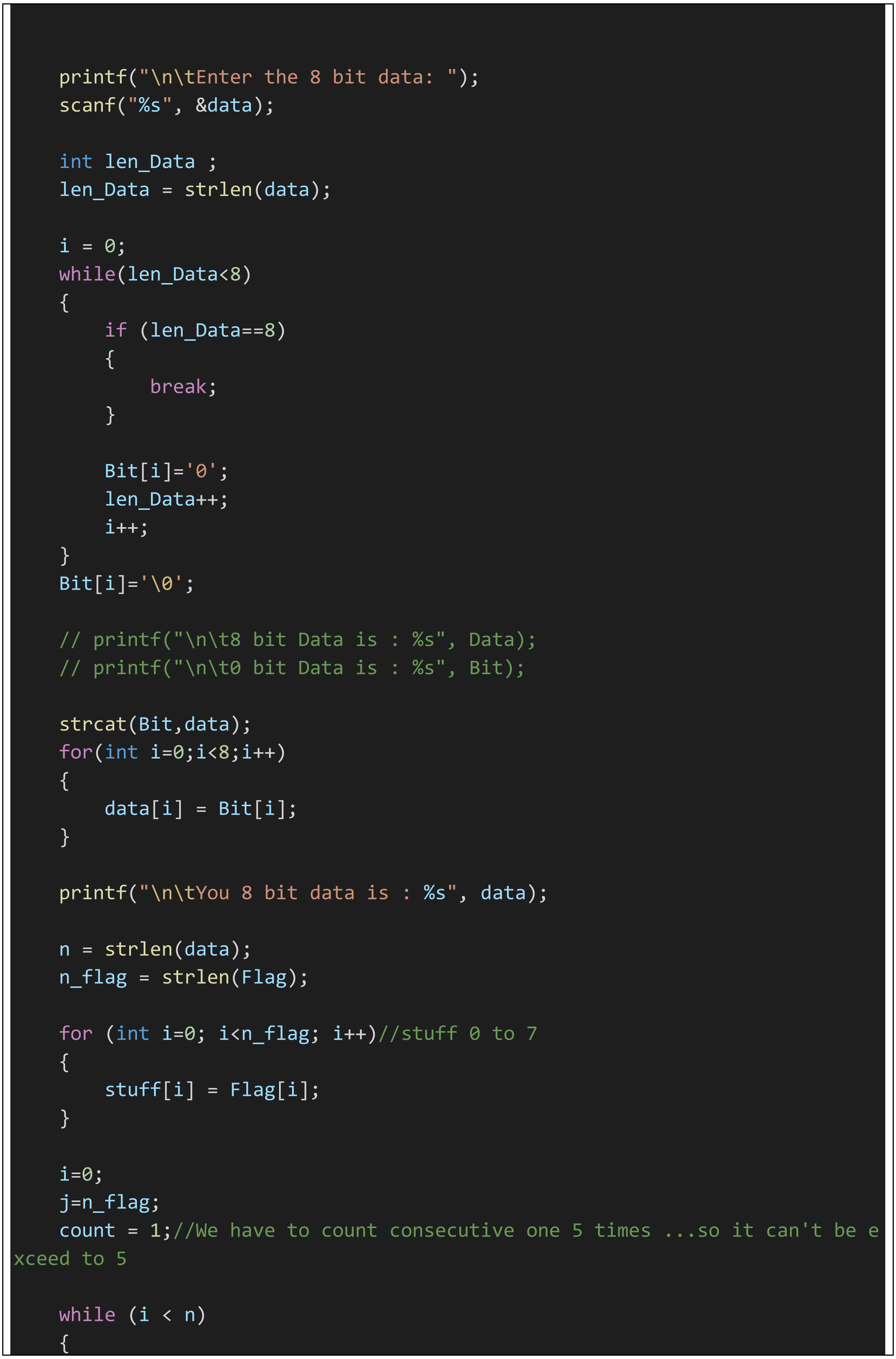


```
    printf("\n\nStuffed Data from Sender : %s" ,stuff);
    int len_stuff = strlen(stuff);
    int n_flag = strlen(Flag);

    //int i=n_flag;
    int j=0;
    //int count = 1;
    int k;
    //j=0;
    int count = 0;
    for (int i = n_flag; i <(len_stuff - n_flag); i++)
    {
        if (stuff[i] == '1')
        {
            count++;
        }
        else
        {
            count=0 ;
        }
        destuff[j]=stuff[i];
        j++;
        if(count==5 && stuff[i+1]=='0')
        {
            count=0;
            i++;
        }
    }

    printf("\n\nDestuffed Data at receiver: %s" ,destuff);
    printf("\n\nMessage is sent to the client.");

    if (strcmp(destuff , Data) == 0)
    {
        char msg[256] = "Data Received Successfully:)";
        write(clientsocketfd , msg , sizeof(msg));
    }
    else
    {
        char msg[256] = "Data NOT Received Successfully:(";
        write(clientsocketfd , msg , sizeof(msg));
    }

    printf("\n\n");

close(socket_server);
```

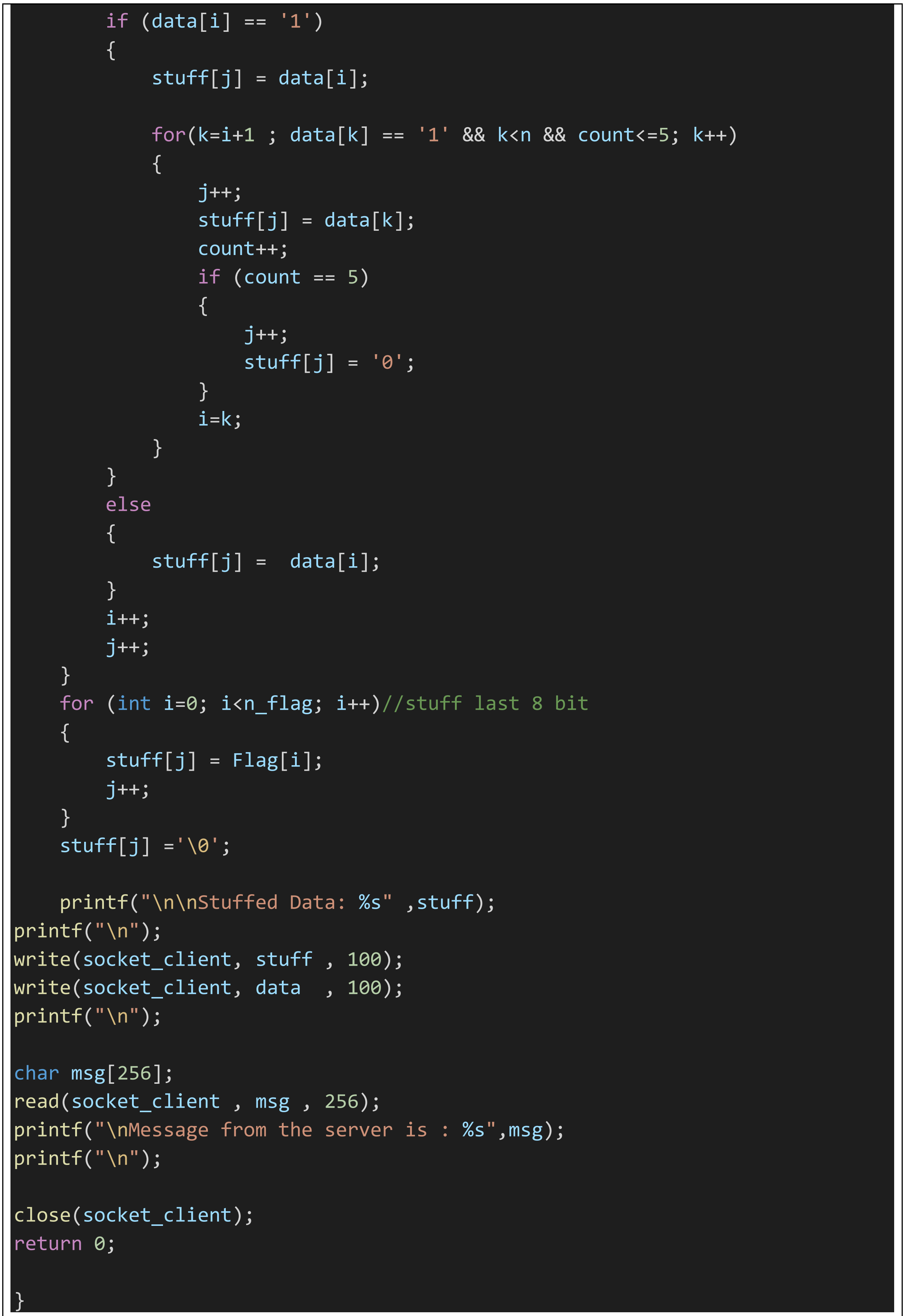
```
return 0;

}
```

Bit_client.c

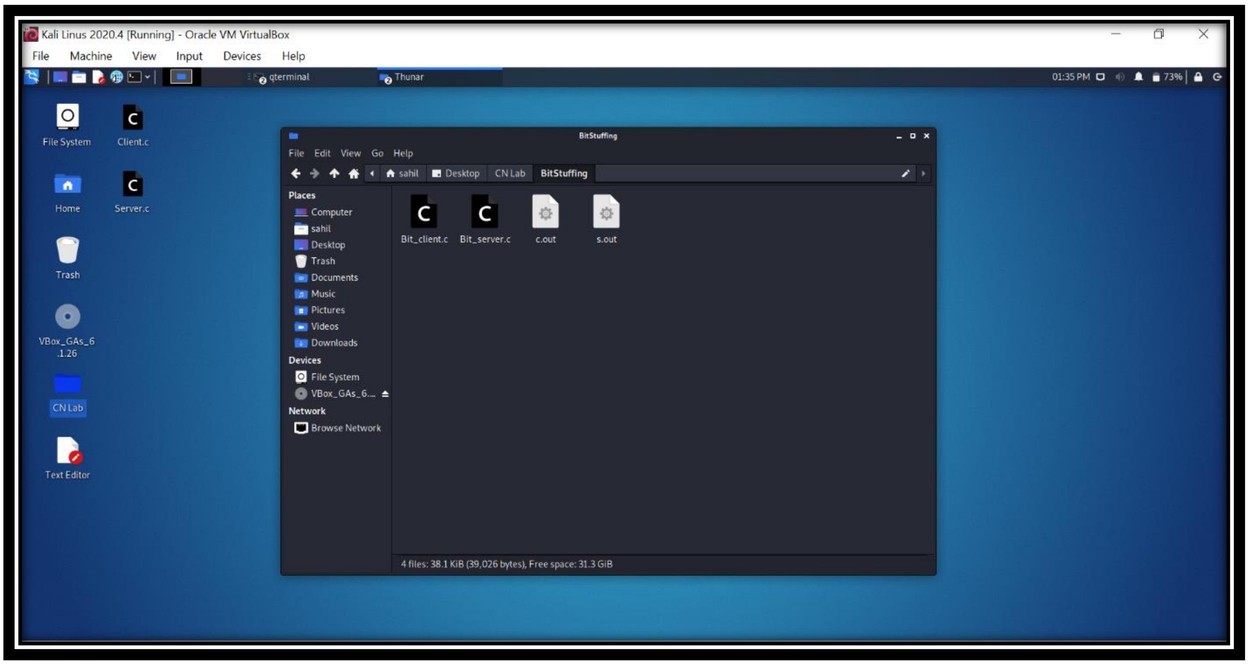
```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<netinet/in.h>

int main()
{

int socket_client , serversocketfd;
struct sockaddr_in serveraddress;
struct hostent *server;

socket_client = socket(AF_INET , SOCK_STREAM , 0);

if(socket_client<0)
printf("Socket is NOT created:(\n");
else
printf("socket is created succesfully:)\n");

serveraddress.sin_family = AF_INET;
serveraddress.sin_port = htons(9000);
serveraddress.sin_addr.s_addr= INADDR_ANY;


int connectionstatus = connect(socket_client,
                              (struct sockaddr *) &serveraddress,
                               sizeof(serveraddress));

if(connectionstatus == -1)
{
    printf("There was an error in the connection with server:( Try again!\n"
);
}


    char data[100] , stuff[100] , Bit[8];
    int i,j,k,n,n_flag,count;
    char Flag[] = "01111110";
```



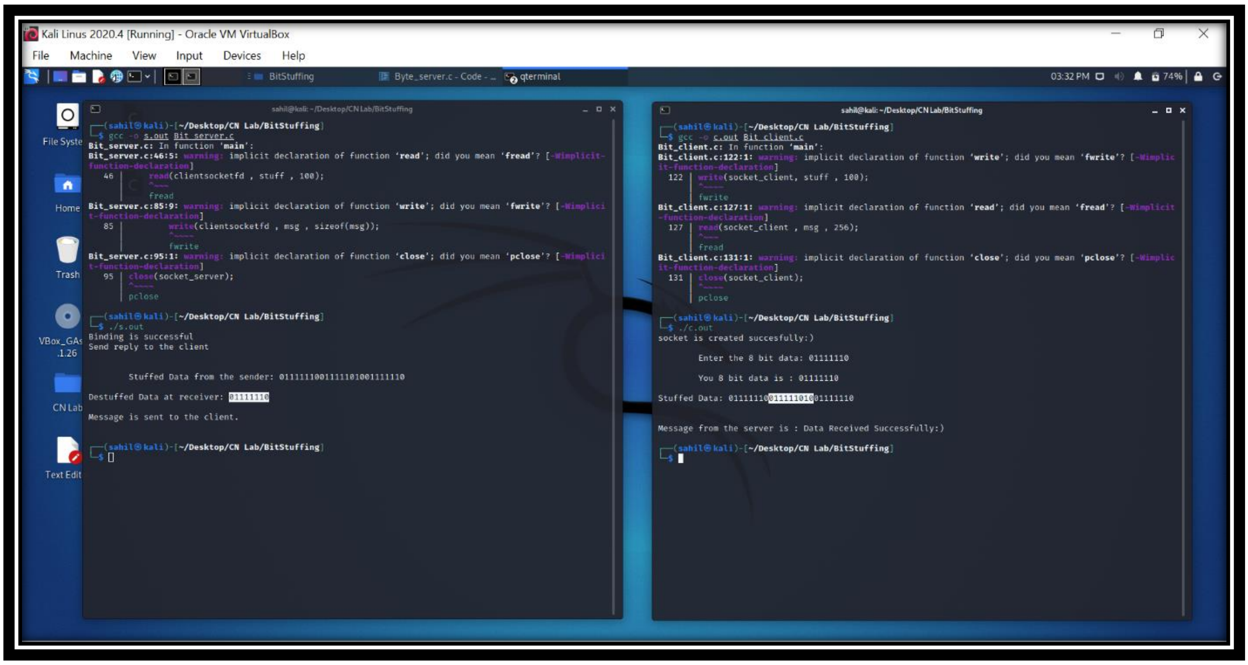
```
    printf("\n\tEnter the 8 bit data: ");
    scanf("%s", &data);

    int len_Data ;
    len_Data = strlen(data);

    i = 0;
    while(len_Data<8)
    {
        if (len_Data==8)
        {
            break;
        }

        Bit[i]='0';
        len_Data++;
        i++;
    }
    Bit[i]='\0';

    // printf("\n\t8 bit Data is : %s", Data);
    // printf("\n\t0 bit Data is : %s", Bit);

    strcat(Bit,data);
    for(int i=0;i<8;i++)
    {
        data[i] = Bit[i];
    }

    printf("\n\tYou 8 bit data is : %s", data);

    n = strlen(data);
    n_flag = strlen(Flag);

    for (int i=0; i<n_flag; i++)//stuff 0 to 7
    {
        stuff[i] = Flag[i];
    }

    i=0;
    j=n_flag;
    count = 1;//We have to count consecutive one 5 times ...so it can't be e
xceed to 5

    while (i < n)
    {
```



```
        if (data[i] == '1')
        {
            stuff[j] = data[i];

            for(k=i+1 ; data[k] == '1' && k<n && count<=5; k++)
            {
                j++;
                stuff[j] = data[k];
                count++;
                if (count == 5)
                {
                    j++;
                    stuff[j] = '0';
                }
                i=k;
            }
        }
        else
        {
            stuff[j] =  data[i];
        }
        i++;
        j++;
    }
    for (int i=0; i<n_flag; i++)//stuff last 8 bit
    {
        stuff[j] = Flag[i];
        j++;
    }
    stuff[j] ='\0';

    printf("\n\nStuffed Data: %s" ,stuff);
printf("\n");
write(socket_client, stuff , 100);
write(socket_client, data  , 100);
printf("\n");

char msg[256];
read(socket_client , msg , 256);
printf("\nMessage from the server is : %s",msg);
printf("\n");

close(socket_client);
return 0;

}
```



4 files: 38.1 KiB (39,026 bytes), Free space: 31.3 GiB


Kali Linus 2020.4 [Running] - Oracle VM VirtualBox
File
Machine
View
Input
Devices
Help
BitStuffing
Byte_server.c - Code - ...
qterminal
03:32 PM
74%
sahil@kali: ~/Desktop/CN Lab/BitStuffing
sahil@kali: ~/Desktop/CN Lab/BitStuffing

```
┌──(sahil㉿kali)-[~/Desktop/CN Lab/BitStuffing]
└─$ gcc -o s.out Bit_server.c
Bit_server.c: In function 'main':
Bit_server.c:46:5: warning: implicit declaration of function 'read'; did you mean 'fread'? [-Wimplicit-function-declaration]
   46 |     read(clientsocketfd , stuff , 100);
      |     ^~~~
      |     fread
Bit_server.c:85:9: warning: implicit declaration of function 'write'; did you mean 'fwrite'? [-Wimplicit-function-declaration]
   85 |         write(clientsocketfd , msg , sizeof(msg));
      |         ^~~~~
      |         fwrite
Bit_server.c:95:1: warning: implicit declaration of function 'close'; did you mean 'pclose'? [-Wimplicit-function-declaration]
   95 | close(socket_server);
      | ^~~~~
      | pclose

┌──(sahil㉿kali)-[~/Desktop/CN Lab/BitStuffing]
└─$ ./s.out
Binding is successful
Send reply to the client


        Stuffed Data from the sender: 0111111001111101001111110

Destuffed Data at receiver: 01111110

Message is sent to the client.


┌──(sahil㉿kali)-[~/Desktop/CN Lab/BitStuffing]
└─$ 
```

```
┌──(sahil㉿kali)-[~/Desktop/CN Lab/BitStuffing]
└─$ gcc -o c.out Bit_client.c
Bit_client.c: In function 'main':
Bit_client.c:122:1: warning: implicit declaration of function 'write'; did you mean 'fwrite'? [-Wimplicit-function-declaration]
  122 | write(socket_client, stuff , 100);
      | ^~~~~
      | fwrite
Bit_client.c:127:1: warning: implicit declaration of function 'read'; did you mean 'fread'? [-Wimplicit-function-declaration]
  127 | read(socket_client , msg , 256);
      | ^~~~
      | fread
Bit_client.c:131:1: warning: implicit declaration of function 'close'; did you mean 'pclose'? [-Wimplicit-function-declaration]
  131 | close(socket_client);
      | ^~~~~
      | pclose

┌──(sahil㉿kali)-[~/Desktop/CN Lab/BitStuffing]
└─$ ./c.out
socket is created succesfully:)

        Enter the 8 bit data: 01111110

        You 8 bit data is : 01111110

Stuffed Data: 0111111001111101001111110


Message from the server is : Data Received Successfully:)

┌──(sahil㉿kali)-[~/Desktop/CN Lab/BitStuffing]
└─$ 
```