Network and Communication
Lab Assessment-1

BY: RIYA SHRESTHA

17BCE2346

Q. Error Detection and Error Correction Mechanism

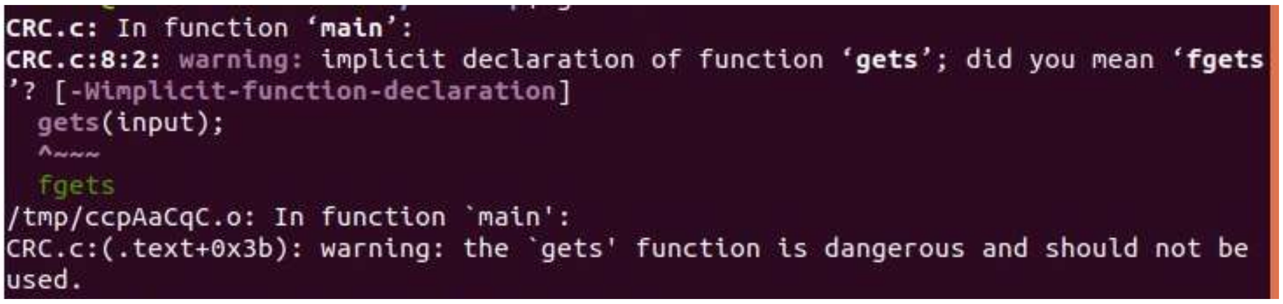
a. CRC(CYCLIC REDUNDANCY CHECK):

CODE:

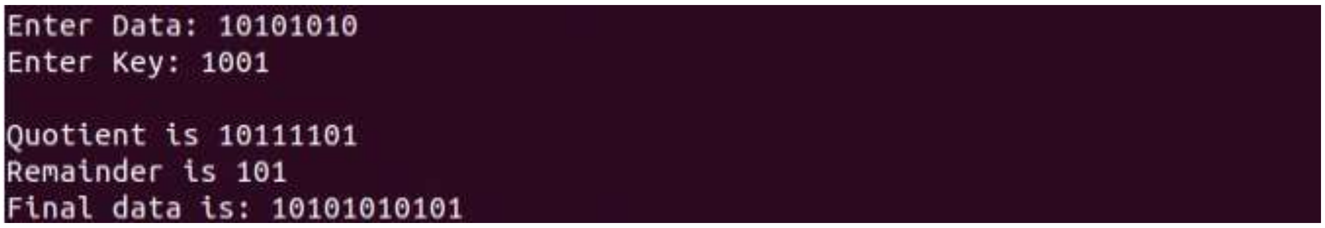
```
#include<stdio.h>
#include<string.h>
void main()
{
	int i,j,keylen,msglen;
	char input[100], key[30],temp[30],quot[100],rem[30],key1[30];
	printf("Enter Data: ");
	gets(input);
	printf("Enter Key: ");
	gets(key);
	keylen=strlen(key);
	msglen=strlen(input);
	strcpy(key1,key);
	for (i=0;i<keylen-1;i++)
	{
		input[msglen+i]='0';
	}
	for (i=0;i<keylen;i++)
	 temp[i]=input[i];
	for (i=0;i<msglen;i++)
   {
		quot[i]=temp[0];
		if(quot[i]=='0')
		{
			for(j=0;j<keylen;j++)
				key[j]='0';
		}
		else

		 for (j=0;j<keylen;j++)
			key[j]=key1[j];
		for (j=keylen-1;j>0;j--)
		{
			if(temp[j]==key[j])
			 rem[j-1]='0';
			else
			 rem[j-1]='1';
		}
		rem[keylen-1]=input[i+keylen];
		strcpy(temp,rem);
	}
	strcpy(rem,temp);
	printf("\nQuotient is ");
	for (i=0;i<msglen;i++)
	printf("%c",quot[i]);
	printf("\nRemainder is ");
	for (i=0;i<keylen-1;i++)
```

```
             printf("%c",rem[i]);
            printf("\nFinal data is: ");
            for (i=0;i<msglen;i++)
            printf("%c",input[i]);
            for (i=0;i<keylen-1;i++)
            printf("%c",rem[i]);
            printf("\n");
}
```

OUTPUT:



```
CRC.c: In function 'main':
CRC.c:8:2: warning: implicit declaration of function 'gets'; did you mean 'fgets
'? [-Wimplicit-function-declaration]
  gets(input);
  ^~~~
  fgets
/tmp/ccpAaCqC.o: In function `main':
CRC.c:(.text+0x3b): warning: the `gets' function is dangerous and should not be
used.
```



```
Enter Data: 10101010
Enter Key: 1001

Quotient is 10111101
Remainder is 101
Final data is: 10101010101
```

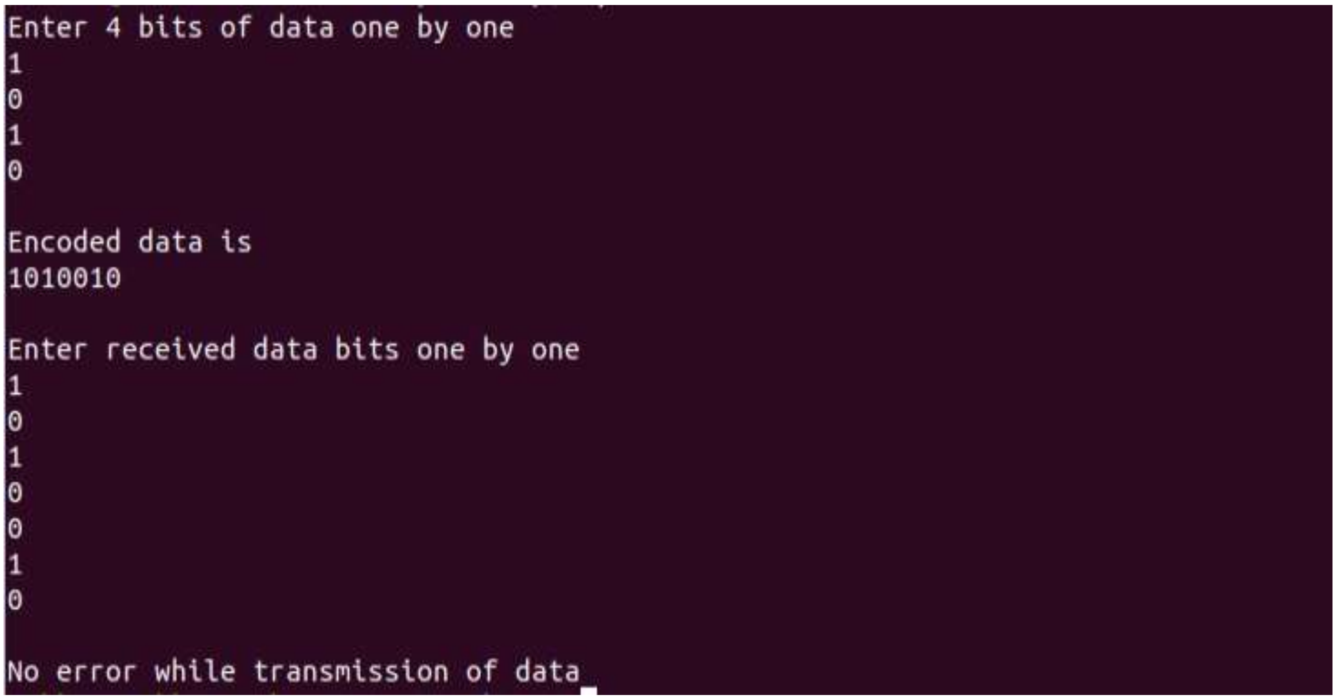
b. Hamming Code:

CODE:

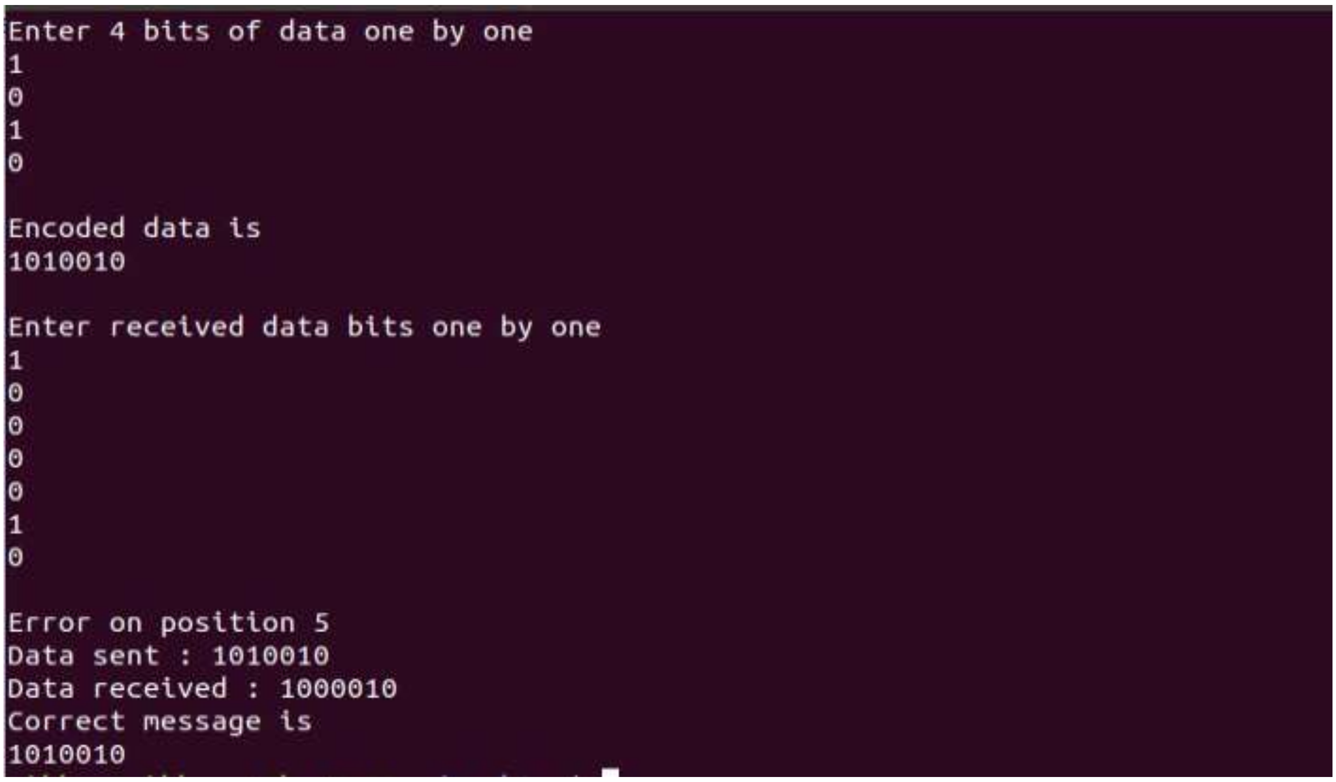
```
#include<stdio.h>
void main()
{
            int data[10];
   int dataatrec[10],c,c1,c2,c3,i;
   printf("Enter 4 bits of data one by one\n");
   scanf("%d",&data[0]);
   scanf("%d",&data[1]);
   scanf("%d",&data[2]);
   scanf("%d",&data[4]);

   //Calculation of even parity
   data[6]=data[0]^data[2]^data[4];
            data[5]=data[0]^data[1]^data[4];
            data[3]=data[0]^data[1]^data[2];
```

```
        printf("\nEncoded data is\n");
        for(i=0;i<7;i++)
        printf("%d",data[i]);

   printf("\n\nEnter received data bits one by one\n");
   for(i=0;i<7;i++)
        scanf("%d",&dataatrec[i]);

   c1=dataatrec[6]^dataatrec[4]^dataatrec[2]^dataatrec[0];
        c2=dataatrec[5]^dataatrec[4]^dataatrec[1]^dataatrec[0];
        c3=dataatrec[3]^dataatrec[2]^dataatrec[1]^dataatrec[0];
        c=c3*4+c2*2+c1 ;
   if(c==0)
        {
                printf("\nNo error while transmission of data\n");
   }
        else
        {
                printf("\nError on position %d",c);

                printf("\nData sent : ");
    for(i=0;i<7;i++)
        printf("%d",data[i]);

                printf("\nData received : ");
    for(i=0;i<7;i++)
    printf("%d",dataatrec[i]);

                printf("\nCorrect message is\n");
                //if errorneous bit is 0 we complement it else vice versa
                if(dataatrec[7-c]==0)
                        dataatrec[7-c]=1;
    else
                        dataatrec[7-c]=0;

                for(i=0;i<7;i++)
                {
                        printf("%d",dataatrec[i]);
                }
                printf("\n");
        }
}
```

OUTPUT:



```
Enter 4 bits of data one by one
1
0
1
0

Encoded data is
1010010

Enter received data bits one by one
1
0
1
0
0
1
0

No error while transmission of data
```



```
Enter 4 bits of data one by one
1
0
1
0

Encoded data is
1010010

Enter received data bits one by one
1
0
0
0
0
1
0

Error on position 5
Data sent : 1010010
Data received : 1000010
Correct message is
1010010
```

Q. Flow control mechanisms

a. Go Back N ARQ:(Using Sockets)

Server Side:

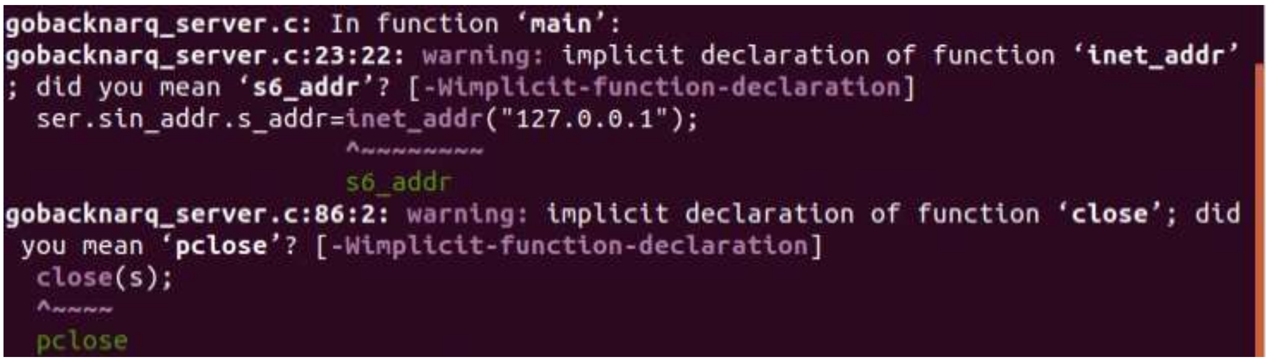
CODE:

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

#define W 5

char a[10];
char b[10];
void alpha9(int);

int main()
{
        int s, f, w1, c=1, x, i=0, j, n, p=0, e=0; struct
        sockaddr_in ser;
        s=socket(AF_INET,SOCK_STREAM,0);
        ser.sin_family=AF_INET;
        ser.sin_port=6500;
        ser.sin_addr.s_addr=inet_addr("127.0.0.1");
        connect(s,(struct sockaddr *) &ser, sizeof(ser));
        printf("\nTCP connection Established.\n");
        printf("\nEnter the number of Frames: ");
        scanf("%d", &f);
        alpha9(f);
        send(s,a,sizeof(a),0);strcpy(b,"Time out");
        while(1)
        {
                for(i=0;i<W;i++)
                {
                        alpha9(c);
                        send(s,a,sizeof(a),0);
                        if(c<=f)
                        {
                                printf("\nFrame %d Sent", c);
                                c++;
                        }
                }
                i=0;

                w1=W;
                while(i<W)
                {
                        recv(s,a,sizeof(a),0);
                        p=atoi(a);
                        if(strcmp(a,b)==0)
                        {
```

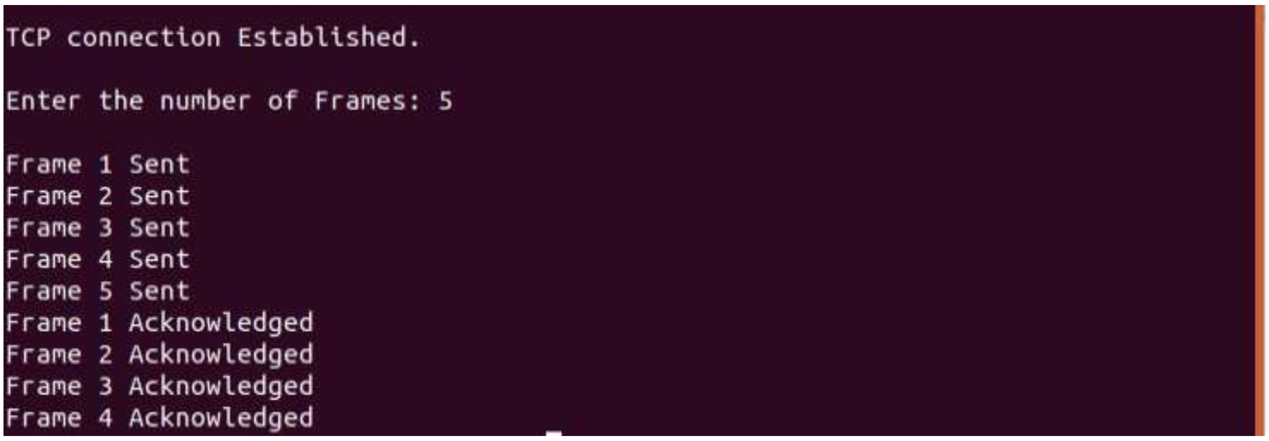
```
                        e=c-w1;
                        if(e<f)
                        {
                                printf("\nTime Out, Resent Frame %d onwards", e);
                        }
                        break;
                }
                else
                {

                        if(p<=f)
                        {
                                printf("\nFrame %s Acknowledged", a);
                                w1--;
                        }
                        else
                        {
                                break;
                        }
                }
                if(p>f)
                {
                        break;
                }
                i++;
        }

        if(w1==0 && c>f)
        {
                send(s,b,sizeof(b),0);
                break;
        }
        else
        {
                c=c-w1;
                w1=W;
        }
    }

    close(s);
    return 0;
}

void alpha9(int z)
{
    int k, i=0, j, g;
    k=z;
    while(k>0)
    {
        i++;
        k=k/10;
    }
    g=i;
    i--;

    while(z>0)
    {
        k=z%10;
        a[i]=k+48;
        i--;
        z=z/10;
```

```
	}
	a[g]='\0';
}
```

OUTPUT:



```
gobacknarq_server.c: In function ‘main’:
gobacknarq_server.c:23:22: warning: implicit declaration of function ‘inet_addr’
; did you mean ‘s6_addr’? [-Wimplicit-function-declaration]
  ser.sin_addr.s_addr=inet_addr("127.0.0.1");
                      ^~~~~~~~~
                      s6_addr
gobacknarq_server.c:86:2: warning: implicit declaration of function ‘close’; did
 you mean ‘pclose’? [-Wimplicit-function-declaration]
  close(s);
  ^~~~~
  pclose
```



```
TCP connection Established.

Enter the number of Frames: 5

Frame 1 Sent
Frame 2 Sent
Frame 3 Sent
Frame 4 Sent
Frame 5 Sent
Frame 1 Acknowledged
Frame 2 Acknowledged
Frame 3 Acknowledged
Frame 4 Acknowledged
```

Client Side:

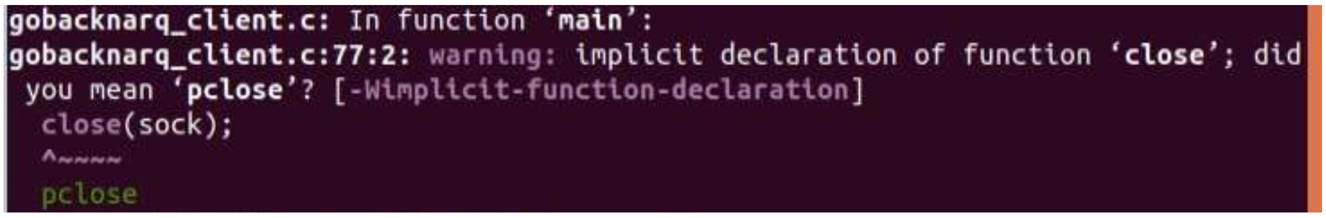
CODE:

```
#include<stdio.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<string.h>
#include<time.h>
#include<stdlib.h>
#include<ctype.h>
#include<arpa/inet.h>
#define W 5
#define P1 50
#define P2 10
char a[10];
char b[10];

void alpha9(int);

int main()
{
	struct sockaddr_in ser,cli;
	int s,n,sock,i,j,c=1,f;
```

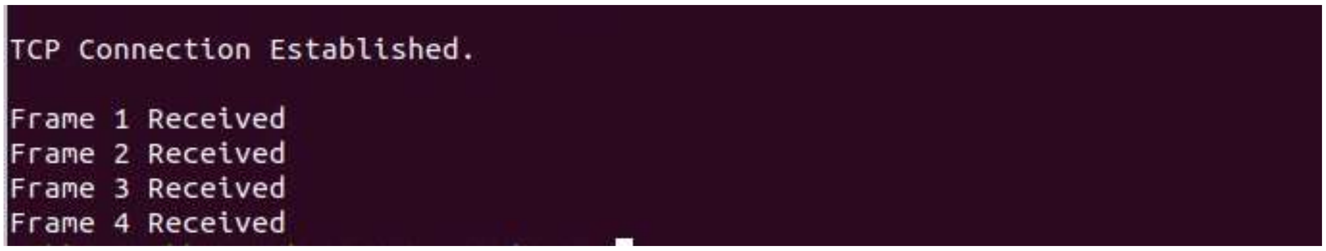
```
unsigned int s1;
s=socket(AF_INET,SOCK_STREAM,0);
ser.sin_family=AF_INET;
ser.sin_port=6500;
ser.sin_addr.s_addr=inet_addr("127.0.0.1");
bind(s,(struct sockaddr *) &ser, sizeof(ser));
listen(s,1);
n=sizeof(cli);
sock=accept(s,(struct sockaddr *)&cli, &n);
printf("\nTCP Connection Established.\n");
s1=(unsigned int) time(NULL);
srand(s1);
strcpy(b,"Time Out ");
recv(sock,a,sizeof(a),0);
f=atoi(a);
while(1)
{
        for(i=0;i<W;i++)
        {
                recv(sock,a,sizeof(a),0);
                if(strcmp(a,b)==0)
                {
                        break;
                }
        }
        i=0;

        while(i<W)
        {
                j=rand()%P1;
                if(j<P2)
                {
                        send(sock,b,sizeof(b),0);
                        break;
                }
                else
                {
                        alpha9(c);
                        if(c<=f)
                        {
                                printf("\nFrame %s Received ",a);
                                send(sock,a,sizeof(a),0);
                        }
                        else
                        {
                                break;
                        }
                        c++;
                }
                if(c>f)
                {
                        break;
                }
                i++;
        }
}

close(sock);
close(s);
```

```
	return 0;
}

void alpha9(int z)
{
	int k,i=0,j,g;
	k=z;
	while(k>0)
	{
		i++;
		k=k/10;
	}
	g=i;
	i--;

	while(z>0)
	{
		k=z%10;
		a[i]=k+48;
		i--;
		z=z/10;
	}
	a[g]='\0';
}
```

OUTPUT:



```
gobacknarq_client.c: In function ‘main’:
gobacknarq_client.c:77:2: warning: implicit declaration of function ‘close’; did
 you mean ‘pclose’? [-Wimplicit-function-declaration]
  close(sock);
  ^~~~~
  pclose
```



```
TCP Connection Established.

Frame 1 Received
Frame 2 Received
Frame 3 Received
Frame 4 Received
```

b. Selective Repeat ARQ(Using Sockets)

Server Side:

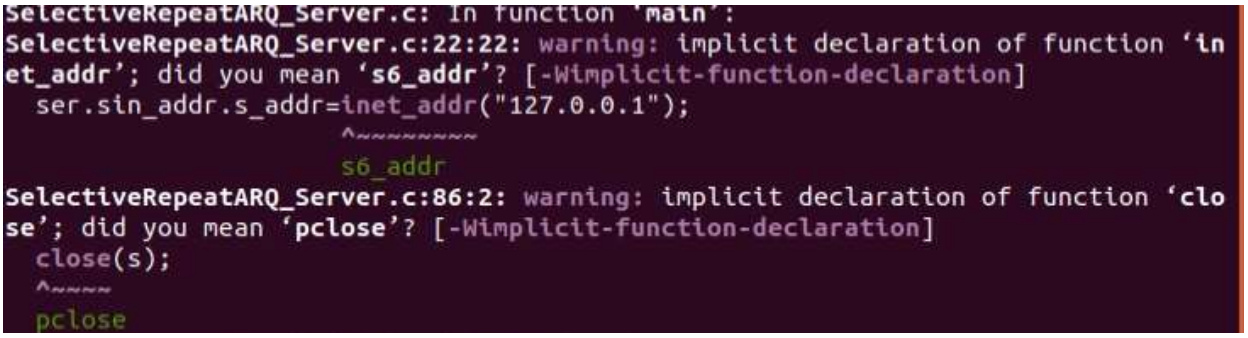
CODE:

```
include<stdio.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<string.h>
#include<time.h>
#include<stdlib.h>
#include<ctype.h>
#define W 5
char a[10];
char b[10];

void alpha9(int);
int con();
int main()
{
        int s,f,wl,c=1,x,i=0,j,n,p=0,e=0; struct
        sockaddr_in ser;
        s=socket(AF_INET,SOCK_STREAM,0);
        ser.sin_family=AF_INET;
        ser.sin_port=6500;
        ser.sin_addr.s_addr=inet_addr("127.0.0.1");
        connect(s,(struct sockaddr *) &ser, sizeof(ser));
        printf("\nTCP Connection Established.\n");
        printf("\nEnter the number of Frames: ");
        scanf("%d",&f);
        alpha9(f);
        send(s,a,sizeof(a),0);
        strcpy(b,"Time Out ");
        while(1)
        {
                for(i=0;i<W;i++)
                {
                        alpha9(c);
                        send(s,a,sizeof(a),0);
                        if(c<=f)
                        {
                                printf("\nFrame %d Sent",c);
                                c++;
                        }
                }
                i=0;wl=W;
                while(i<W)
                {
                        recv(s,a,sizeof(a),0);
                        p=atoi(a);
                        if(a[0]=='N')
                        {
```

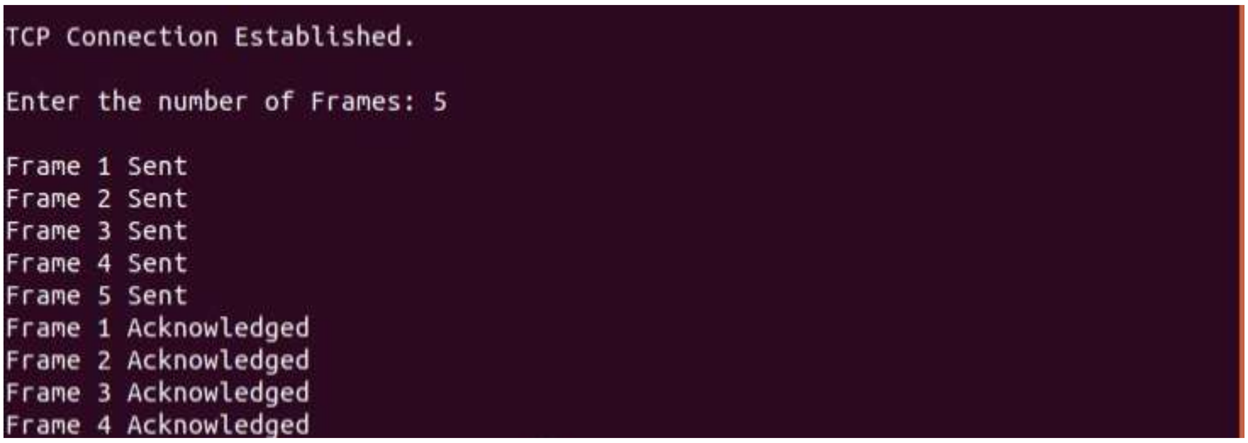
```
				e=con();
				if(e<f)
				{
					printf("\nNAK %d",e);
					printf("\nFrame %d sent",e);
					i--;
				}
			}
			else
			{

				if(p<=f)
				{
					printf("\nFrame %s Acknowledged",a);
					wl--;
				}
				else
				{
					break;
				}
			}
			if(p>f)
			{
				break;
			}
			i++;
		}

		if(wl==0 && c>f)
		{
			send(s,b,sizeof(b),0);
			break;
		}
		else
		{
			c=c-wl;
			wl=W;
		}
	}

	close(s);
	return 0;
}

void alpha9(int z)
{
	int k,i=0,j,g;
	k=z;
	while(k>0)
	{
		i++;
		k=k/10;
	}
	g=i;
	i--;

	while(z>0)
	{
		k=z%10;
		a[i]=k+48;
		i--;
```

```
                z=z/10;
        }
        a[g]='\0';
}

int con()
{
        char k[9]; int
        i=1;
        while(a[i]!='\0')
        {
                k[i-1]=a[i];
                i++;
        }
        k[i-1]='\0';
        i=atoi(k);
        return i;
}
```

OUTPUT:



```
SelectiveRepeatARQ_Server.c: In function 'main':
SelectiveRepeatARQ_Server.c:22:22: warning: implicit declaration of function 'in
et_addr'; did you mean 's6_addr'? [-Wimplicit-function-declaration]
  ser.sin_addr.s_addr=inet_addr("127.0.0.1");
                      ^~~~~~~~~
                      s6_addr
SelectiveRepeatARQ_Server.c:86:2: warning: implicit declaration of function 'clo
se'; did you mean 'pclose'? [-Wimplicit-function-declaration]
  close(s);
  ^~~~~
  pclose
```



```
TCP Connection Established.

Enter the number of Frames: 5

Frame 1 Sent
Frame 2 Sent
Frame 3 Sent
Frame 4 Sent
Frame 5 Sent
Frame 1 Acknowledged
Frame 2 Acknowledged
Frame 3 Acknowledged
Frame 4 Acknowledged
```

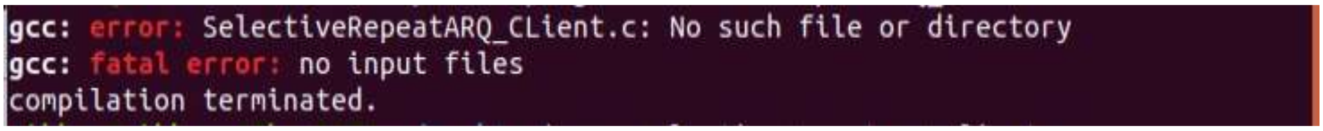
Client Side:

CODE:

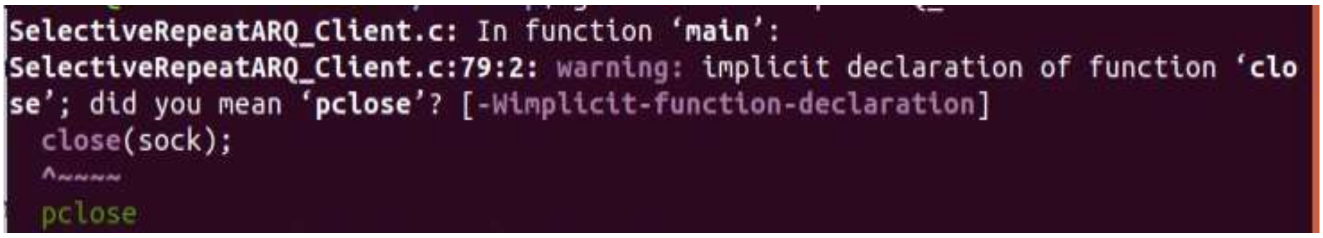
```
#include<stdio.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<string.h>
```

```
#include<time.h>
#include<stdlib.h>
#include<ctype.h>
#include<arpa/inet.h>
#define W 5
#define P1 50
#define P2 10
char a[10];
char b[10];
void alpha9(int);
void alp(int);

int main()
{
        struct sockaddr_in ser,cli;
        int s,n,sock,i,j,c=1,f;
        unsigned int s1;
        s=socket(AF_INET,SOCK_STREAM,0);
        ser.sin_family=AF_INET;
        ser.sin_port=6500;
        ser.sin_addr.s_addr=inet_addr("127.0.0.1");
        bind(s,(struct sockaddr *) &ser, sizeof(ser));
        listen(s,1);
        n=sizeof(cli);
        sock=accept(s,(struct sockaddr *)&cli, &n);
        printf("\nTCP Connection Established.\n");
        s1=(unsigned int) time(NULL);
        srand(s1);
        strcpy(b,"Time Out ");
        recv(sock,a,sizeof(a),0);
        f=atoi(a);
        while(1)
        {
                for(i=0;i<W;i++)
                {
                        recv(sock,a,sizeof(a),0);
                        if(strcmp(a,b)==0)
                        {
                                break;
                        }
                }
                i=0;

                while(i<W)
                {
                        L:
                                j=rand()%P1;
                        if(j<P2)
                        {
                                alp(c);
                                send(sock,b,sizeof(b),0);
                                goto L;
                        }
                        else
                        {
                                alpha9(c);
                                if(c<=f)
                                {
                                        printf("\nFrame %s Received ",a);
                                        send(sock,a,sizeof(a),0);
                                }
                                else
                                {
                                        break;
                                }
                                c++;
                        }
                        if(c>f)
                        {
                                break;
                        }
                        i++;
                }
        }

        close(sock);
        close(s);
        return 0;
}

void alpha9(int z)
{
        int k,i=0,j,g;
        k=z;
        while(k>0)
        {
                i++;
                k=k/10;
        }
        g=i;
        i--;

        while(z>0)
        {
                k=z%10;
                a[i]=k+48;
                i--;
                z=z/10;
        }
        a[g]='\0';
}

void alp(int z)
{
        int k,i=1,j,g;
        k=z;
        b[0]='N';
        while(k>0)
        {
                i++;
                k=k/10;
        }
        g=i;
        i--;

        while(z>0)
        {
                k=z%10;
                b[i]=k+48;
                i--;
                z=z/10;
        }
        b[g]='\0';
}
```

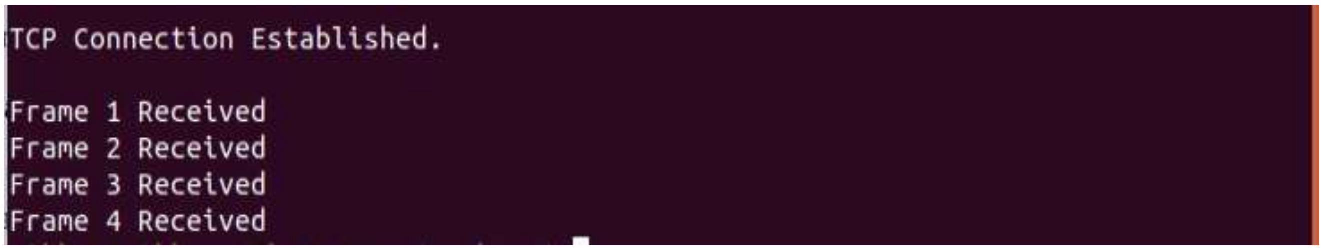
OUTPUT:



```
gcc: error: SelectiveRepeatARQ_CLient.c: No such file or directory
gcc: fatal error: no input files
compilation terminated.
```



```
SelectiveRepeatARQ_Client.c: In function ‘main’:
SelectiveRepeatARQ_Client.c:79:2: warning: implicit declaration of function ‘clo
se’; did you mean ‘pclose’? [-Wimplicit-function-declaration]
  close(sock);
  ^~~~~
  pclose
```



```
TCP Connection Established.

Frame 1 Received
Frame 2 Received
Frame 3 Received
Frame 4 Received
```

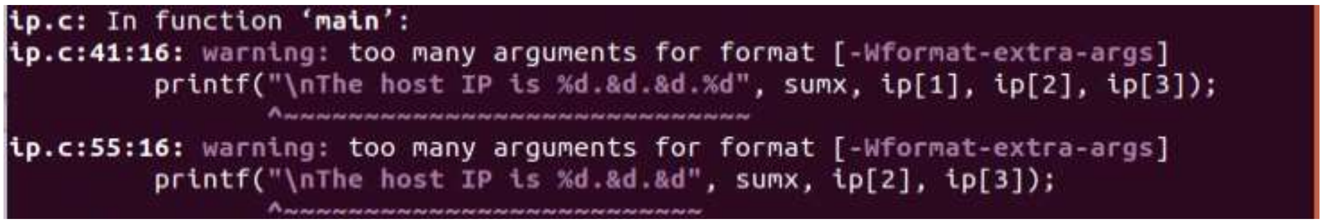
Q. IP addressing – Classless addressing(Calculation, Validation of IP

Addresses) CODE:

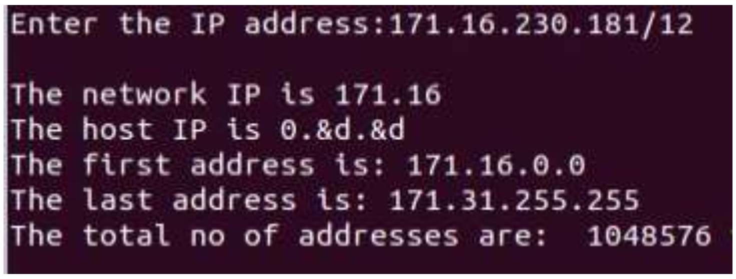
```
#include<stdio.h>
#include<math.h>
#include<stdlib.h>

int main(){
   int ip[4] ,n ,i ,j ,s ,t ,sum;
        printf("Enter the IP address:");
   scanf("%d.%d.%d.%d/%d", &ip[0], &ip[1], &ip[2], &ip[3], &n);
        for(int z=0; z<4;z++)
        {
                if(ip[z]>255)
                {
                        printf("THE IP ADDRESS IS INVALID! ABORTING!!!\n");
                        exit(0);
```

```
                }
        }
        if(n>30)
        {
                        printf("SUBNET MASK MAX LIMIT EXCEEDED! ABORTING!!!\n");
                        exit(0);
        }
int binary[4][8];

for(i = 0; i < 4; i++){
    s = ip[i];
    for(t = 0; t < 8; t++){
        binary[i][7-t] = s % 2;
        s = s / 2;
    }
}
if(n < 9){
    int sumx = 0;
    for(int i = 0; i < n; i++){
        sumx += binary[0][i] * pow(2, (7-i));
    }
    printf("\nThe network IP is %d", sumx);
    sumx = 0;
    for(int i = n; i < 8; i++){
        sumx += binary[0][i] * pow(2, (7-i));
    }
    printf("\nThe host IP is %d.&d.&d.%d", sumx, ip[1], ip[2], ip[3]);
}
else if(n >= 9 && n < 17){
    n -= 8;
    int sumx = 0;
    for(int i = 0; i < n; i++){
        sumx += binary[1][i] * pow(2, (7-i));
    }
    printf("\nThe network IP is %d.%d", ip[0], sumx);
    sumx = 0;
    for(int i = n; i < 8; i++){
        sumx += binary[1][i] * pow(2, (7-i));
    }
    n += 8;
    printf("\nThe host IP is %d.&d.&d", sumx, ip[2], ip[3]);
}
else if(n >= 17 && n < 25){
    n -= 16;
    int sumx = 0;
    for(int i = 0; i < n; i++){
        sumx += binary[2][i] * pow(2, (7-i));
    }
    printf("\nThe network IP is %d.%d.%d", ip[0], ip[1], sumx);
    sumx = 0;
    for(int i = n; i < 8; i++){
        sumx += binary[2][i] * pow(2, (7-i));
    }
    n += 16;
    printf("\nThe host IP is %d.%d", sumx, ip[3]);
}
else if(n >= 25 && n < 33){
        n -= 24;
        int sumx = 0;
        for(int i = 0; i < n; i++){
            sumx += binary[3][i] * pow(2, (7-i));
        }
        printf("\nThe network IP is %d.%d.%d.%d", ip[0], ip[1], ip[2], sumx);
        sumx = 0;
        for(int i = n; i < 8; i++){
            sumx += binary[3][i] * pow(2, (7-i));
        }
        n += 24;
        printf("\nThe host IP is %d", sumx);
    }
    else{
        printf("Wrong choice for mask");
        return 0;
    }
    int k = 32 - n;
    j = 0, t = 0;
    for(i = 0; i < k; i++){
        if(j != 8){
            binary[3-t][7-j] = 0;
            j++;
        }
        else{
            j = 0;
            t++;
            binary[3-t][7-j] = 0;
            j++;
        }
    }
    int first_addr[4], last_addr[4];
    for(i = 0; i < 4; i++){
        sum = 0;
        for(j = 0; j < 8; j++){
            sum += binary[i][7-j] * (pow(2, j));
        }
        first_addr[i] = sum;
    }
    printf("\nThe first address is: ");
    for(i = 0; i < 3; i++){
        printf("%d.", first_addr[i]);
    }
    printf("%d", first_addr[3]);
    for(i = 0;i < 4; i++){
        s = ip[i];
        for(t = 0; t < 8; t++){
            binary[i][7-t] = s % 2;
            s = s / 2;
        }
    }
    j = 0, t = 0;
    for(i = 0; i < k; i++){
        if(j != 8){
            binary[3-t][7-j] = 1;
            j++;
        }
        else{
            j = 0;
            t++;
            binary[3-t][7-j] = 1;
            j++;
        }
    }
    for(i = 0; i < 4; i++){
        sum = 0;
        for(j = 0; j < 8; j++){
            sum += binary[i][7-j] * pow(2, j);
        }
        last_addr[i] = sum;
    }
    printf("\nThe last address is: ");
    for(i = 0; i < 3; i++){
        printf("%d.", last_addr[i]);
    }
    printf("%d", last_addr[3]);
    n = pow(2, k);
    printf("\nThe total no of addresses are:  %d ", n);
}
```

OUTPUT:



```
ip.c: In function ‘main’:
ip.c:41:16: warning: too many arguments for format [-Wformat-extra-args]
         printf("\nThe host IP is %d.&d.&d.%d", sumx, ip[1], ip[2], ip[3]);
                ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
ip.c:55:16: warning: too many arguments for format [-Wformat-extra-args]
         printf("\nThe host IP is %d.&d.&d", sumx, ip[2], ip[3]);
                ^~~~~~~~~~~~~~~~~~~~~~~~~~~~
```



```
Enter the IP address:171.16.230.181/12

The network IP is 171.16
The host IP is 0.&d.&d
The first address is: 171.16.0.0
The last address is: 171.31.255.255
The total no of addresses are:  1048576
```