Q. Write a program in C to implement a simple client-Server application the client will take a data word and division from user and send them to server. The server will find out the code word using CRC and return back to the client. Use unix file socket for the communication.

crc_serv.c

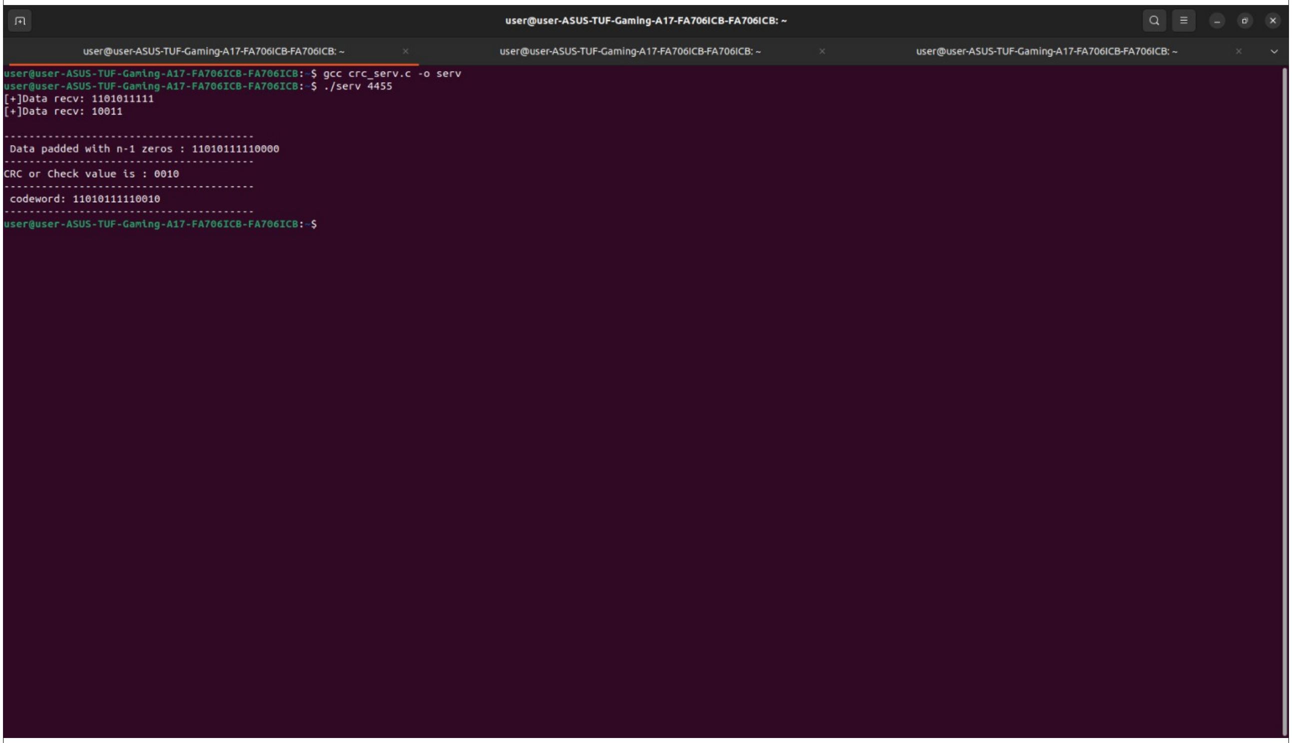
```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#define N strlen(gen_poly)
char data[1024];
char check_value[28];
char gen_poly[10];
int data_length,i,j;

void XOR()
{
for( j = 1;j < N; j++)
check_value[ j] = (( check_value[ j] == gen_poly[ j])?'0':'1');
}

void crc(){
for(i=0;i<N;i++)
check_value[i]=data[i];
do{
if(check_value[0]=='1')
XOR();
for( j=0;j<N-1;j++)
check_value[ j]=check_value[ j+1];
check_value[ j]=data[i++];
}while(i<=data_length+N-1);
}
int main(int argc, char **argv){

  if (argc != 2) {
    printf("Usage: %s <port>\n", argv[0]);
    exit(0);
  }

  char *ip = "127.0.0.1";
  int port = atoi(argv[1]);

  int sockfd;
  struct sockaddr_in server_addr, client_addr;

  socklen_t addr_size;
  int n;
```

```
 sockfd = socket(AF_INET, SOCK_DGRAM, 0);
 if (sockfd < 0) {
   perror("[-]socket error");
   exit(1);
 }

 memset(&server_addr, '\0', sizeof(server_addr));
 server_addr.sin_family = AF_INET;
 server_addr.sin_port = htons(port);
 server_addr.sin_addr.s_addr = inet_addr(ip);

 n = bind(sockfd, (struct sockaddr*)&server_addr, sizeof(server_addr));
 if (n < 0){
   perror("[-]bind error");
   exit(1);
 }

 bzero(data, 1024);
 bzero(gen_poly, 10);
 addr_size = sizeof(client_addr);
 recvfrom(sockfd, data, 1024, 0, (struct sockaddr*)&client_addr, &addr_size);
 recvfrom(sockfd, gen_poly, 1024, 0, (struct sockaddr*)&client_addr, &addr_size);
 printf("[+]Data recv: %s\n", data);
 printf("[+]Data recv: %s\n", gen_poly);
 data_length=strlen(data);
 for(i=data_length;i<data_length+N-1;i++)
data[i]='0';
printf("\n----------------------------------------");
printf("\n Data padded with n-1 zeros : %s",data);
printf("\n----------------------------------------");
crc();
printf("\nCRC or Check value is : %s",check_value);
for(i=data_length;i<data_length+N-1;i++)
data[i]=check_value[i-data_length];
printf("\n----------------------------------------");
printf("\n codeword: %s",data);
printf("\n----------------------------------------\n");
 sendto(sockfd, data, 1024, 0, (struct sockaddr*)&client_addr, sizeof(client_addr));

}
```

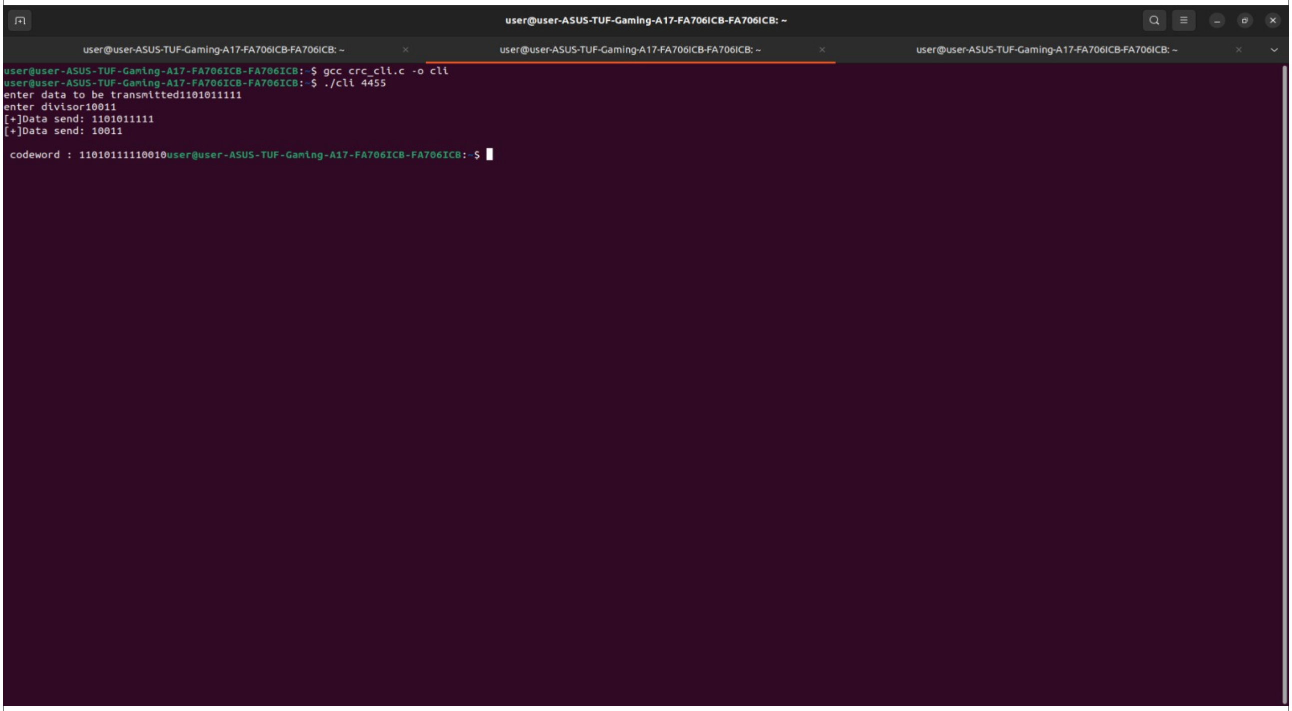
crc_cli.c

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

int main(int argc, char **argv){
  if (argc != 2) {
    printf("Usage: %s <port>\n", argv[0]);
    exit(0);
  }

  char *ip = "127.0.0.1";
  int port = atoi(argv[1]);

  int sockfd;
  struct sockaddr_in addr;
  char buffer[1024],input[1024],divisor[1024];
  char output[1024];
  int t,i;
  socklen_t addr_size;

  sockfd = socket(AF_INET, SOCK_DGRAM, 0);
  memset(&addr, '\0', sizeof(addr));
  addr.sin_family = AF_INET;
  addr.sin_port = htons(port);
  addr.sin_addr.s_addr = inet_addr(ip);
   bzero(input, 1024);
  bzero(divisor, 1024);
  printf("enter data to be transmitted");
  scanf("%s",input);
  printf("enter divisor");
  scanf("%s",divisor);


  strcpy(buffer, input);

  sendto(sockfd, buffer, 1024, 0, (struct sockaddr*)&addr, sizeof(addr));
    sendto(sockfd, divisor, 1024, 0, (struct sockaddr*)&addr, sizeof(addr));
  printf("[+]Data send: %s\n",buffer);
   printf("[+]Data send: %s\n",divisor);
  bzero(buffer, 1024);
  addr_size = sizeof(addr);
  recvfrom(sockfd, output, 1024, 0, (struct sockaddr*)&addr, &addr_size);
  printf("\n codeword : %s",output);

  return 0;
}
```

OUTPUT:-


user@user-ASUS-TUF-Gaming-A17-FA706ICB-FA706ICB: ~
user@user-ASUS-TUF-Gaming-A17-FA706ICB-FA706ICB: ~
user@user-ASUS-TUF-Gaming-A17-FA706ICB-FA706ICB: ~
user@user-ASUS-TUF-Gaming-A17-FA706ICB-FA706ICB: ~

```
user@user-ASUS-TUF-Gaming-A17-FA706ICB-FA706ICB:~$ gcc crc_serv.c -o serv
user@user-ASUS-TUF-Gaming-A17-FA706ICB-FA706ICB:~$ ./serv 4455
[+]Data recv: 1101011111
[+]Data recv: 10011

----------------------------------------
 Data padded with n-1 zeros : 11010111110000
----------------------------------------
CRC or Check value is : 0010
----------------------------------------
 codeword: 11010111110010
----------------------------------------
user@user-ASUS-TUF-Gaming-A17-FA706ICB-FA706ICB:~$
```


user@user-ASUS-TUF-Gaming-A17-FA706ICB-FA706ICB: ~
user@user-ASUS-TUF-Gaming-A17-FA706ICB-FA706ICB: ~
user@user-ASUS-TUF-Gaming-A17-FA706ICB-FA706ICB: ~
user@user-ASUS-TUF-Gaming-A17-FA706ICB-FA706ICB: ~

```
user@user-ASUS-TUF-Gaming-A17-FA706ICB-FA706ICB:~$ gcc crc_cli.c -o cli
user@user-ASUS-TUF-Gaming-A17-FA706ICB-FA706ICB:~$ ./cli 4455
enter data to be transmitted1101011111
enter divisor10011
[+]Data send: 1101011111
[+]Data send: 10011

 codeword : 11010111110010user@user-ASUS-TUF-Gaming-A17-FA706ICB-FA706ICB:~$ 
```