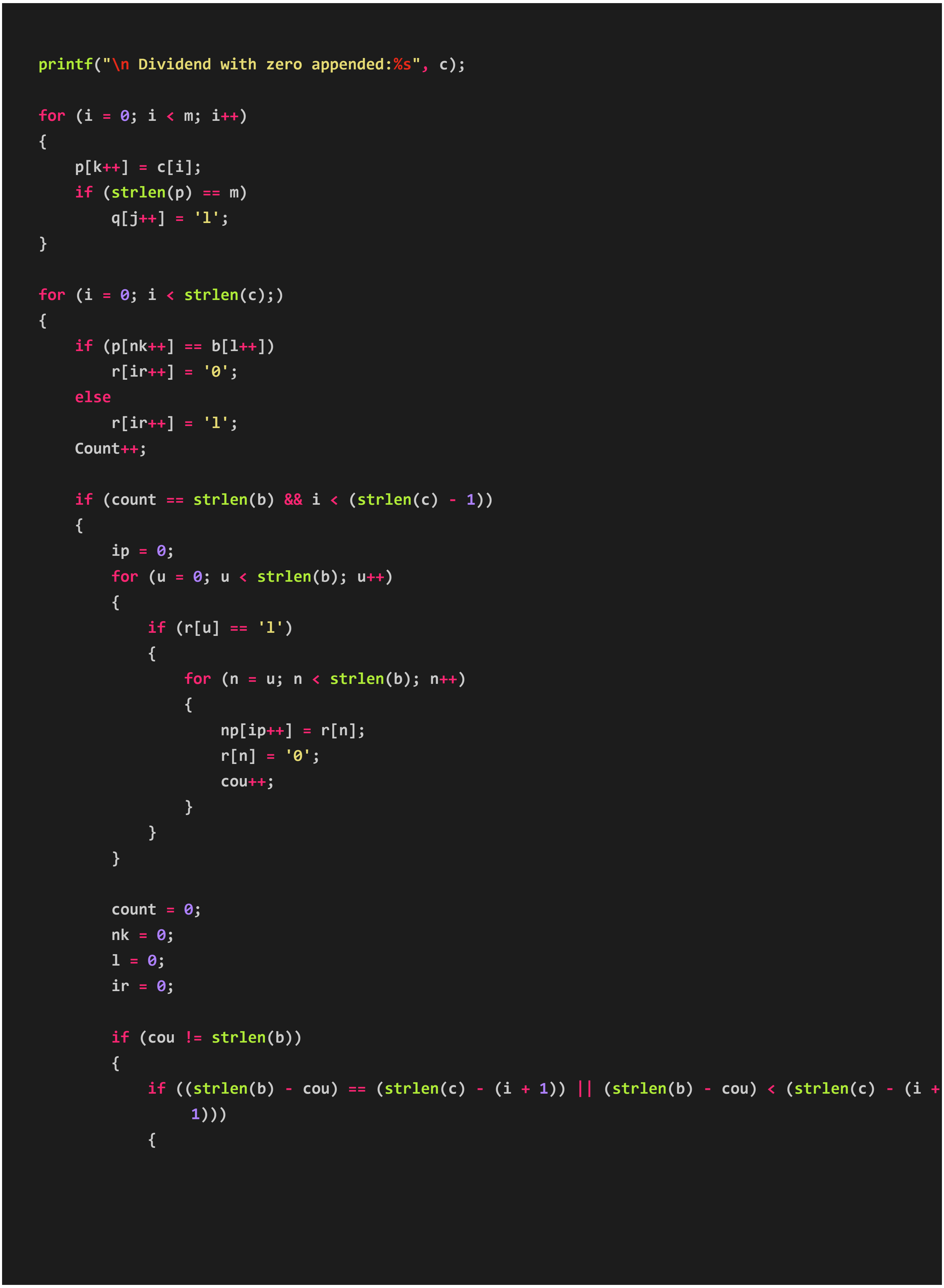
2019103573 **CN LAB - 13** SACHIN RAGHUL T

CYCLIC REDUNDANCY CHECK - CRC

SERVER :

```
#include <stdio.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#define SERV_TCP_PORT 3573

int main(int argc, char **argv)
{
    int sockfd, newsockfd, clength;
    struct sockaddr_in serv_addr, cli_addr;

    char a[30], b[30], c[30] = {0}, q[30] = {0}, p[30] = {0}, np[30] = {0}, crc[10] = {0}, r[30] = {0};
    int n, m, i = 0, j = 0, count = 0, k = 0, l = 0, ir = 0, ip = 0, cou = 0, u = 0, w = 0, nk = 0;

    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    serv_addr.sin_family = AF_INET;
    serv_addr.sin_addr.s_addr = INADDR_ANY;
    serv_addr.sin_port = htons(SERV_TCP_PORT);

    printf("\n Binded...");
    bind(sockfd, (struct sockaddr *)&serv_addr, sizeof(serv_addr));
    listen(sockfd, 5);

    clength = sizeof(cli_addr);
    newsockfd = accept(sockfd, (struct sockaddr *)&cli_addr, &clength);
    read(newsockfd, a, 30);
    read(newsockfd, b, 30);

    m = strlen(b);
    printf("\n Dividend:%s", a);
    printf("\n Divisor:%s", b);
    strcpy(c, a);

    for (i = 0; i < m - 1; i++)
        strcat(c, "0");
```



```
printf("\n Dividend with zero appended:%s", c);

for (i = 0; i < m; i++)
{
    p[k++] = c[i];
    if (strlen(p) == m)
        q[j++] = 'l';
}

for (i = 0; i < strlen(c);)
{
    if (p[nk++] == b[l++])
        r[ir++] = '0';
    else
        r[ir++] = 'l';
    Count++;

    if (count == strlen(b) && i < (strlen(c) - 1))
    {
        ip = 0;
        for (u = 0; u < strlen(b); u++)
        {
            if (r[u] == 'l')
            {
                for (n = u; n < strlen(b); n++)
                {
                    np[ip++] = r[n];
                    r[n] = '0';
                    cou++;
                }
            }
        }

        count = 0;
        nk = 0;
        l = 0;
        ir = 0;

        if (cou != strlen(b))
        {
            if ((strlen(b) - cou) == (strlen(c) - (i + 1)) || (strlen(b) - cou) < (strlen(c) - (i +
                 1)))
            {
                while (cou != strlen(b))
                {
                    i++;
                    np[ip++] = c[i];
                    cou++;
                    W++;

                }
                strcpy(p, np);
                for (u = 0; u < w - 1; u++)
                    q[j++] = '0';
                if (w != 0)
                {
                    i -= strlen(np);
                    w = 0;
                }
            }
            else
            {
                for (; i + 1 < strlen(c);)
                {
                    i++;
                    np[ip++] = c[i];
                    w++;
                }

                if (ip < strlen(b))
                {
                    for (; ip < strlen(b);)
                        np[ip++] = ' ';
                }
                strcpy(r, np);
                for (u = 0; u < w - 1; u++)
                    q[i++] = '0';
                i = strlen(c);
                w = 0;
            }
        }
        if (cou = strlen(b))
        {
            q[j++] = '1';
            cou = 0;
        }
        ip = 0;
        cou = 0;
    }
    i++;
}
```

```
    printf("\n Quotient=%s", q);
    printf("\n Remainder=%s", r);

    for (i = strlen(r) - (m - 1); i <= strlen(r); i++)
        crc[w++] = r[i];

    printf("\n CRC values: %s\n", crc);
    write(newsockfd, q, 30);
    write(newsockfd, r, 30);
    write(newsockfd, crc, 10);
    close(sockfd);

    return 0;
}
```

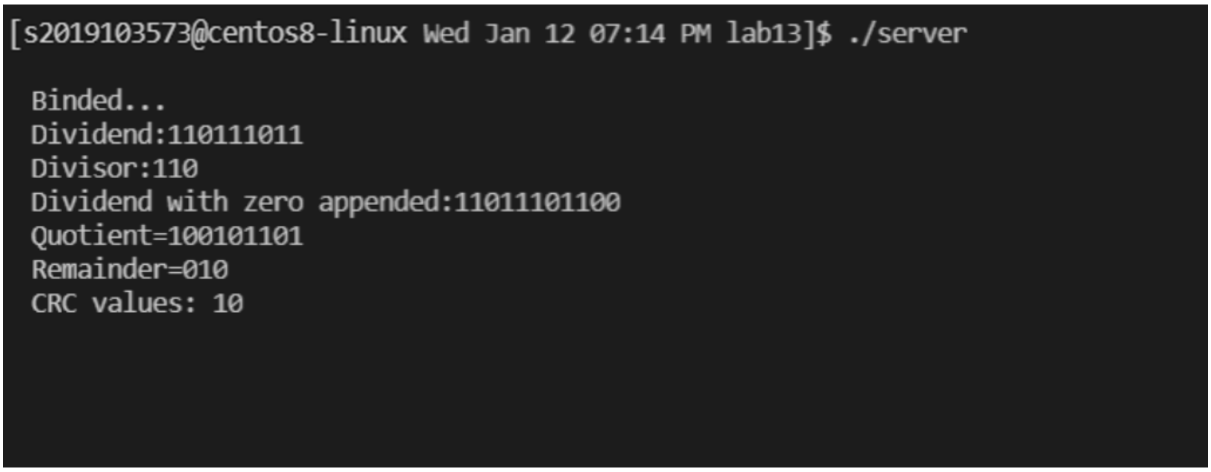
CLIENT :

```
#include<stdio.h>
#include<string.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<netdb.h>
#define SERV_TCP_PORT 3573

int main(int argc,char * * argv)
{
       int sockfd;
       struct sockaddr_in serv_addr;
       struct hostent *server;

       char a[30],b[30],q[30],r[30],crc[10];
       sockfd=socket(AF_INET,SOCK_STREAM,0);

       serv_addr.sin_family=AF_INET;
       serv_addr.sin_addr.s_addr=inet_addr("127.0.0.1");
       serv_addr.sin_port=htons(SERV_TCP_PORT);

       connect(sockfd,(struct sockaddr*)&serv_addr,sizeof(serv_addr));

       printf("\nEnter the dividend:");
       scanf("%s",a);
```

```
	printf("\nEnter the divisor:");
	scanf("%s",b);

	write(sockfd,a,30);
	write(sockfd,b,30);

	printf("\n");
	printf("\nServer result:");

	read(sockfd,q,30);
	read(sockfd,r,30);
	read(sockfd,crc,10);

	printf("\n\nQuotient=%s",q);
	printf("\n\nRemainder=%s",r);
	printf("\n\nCRC values=%s\n",crc);

	close(sockfd);

	return 0;
}
```

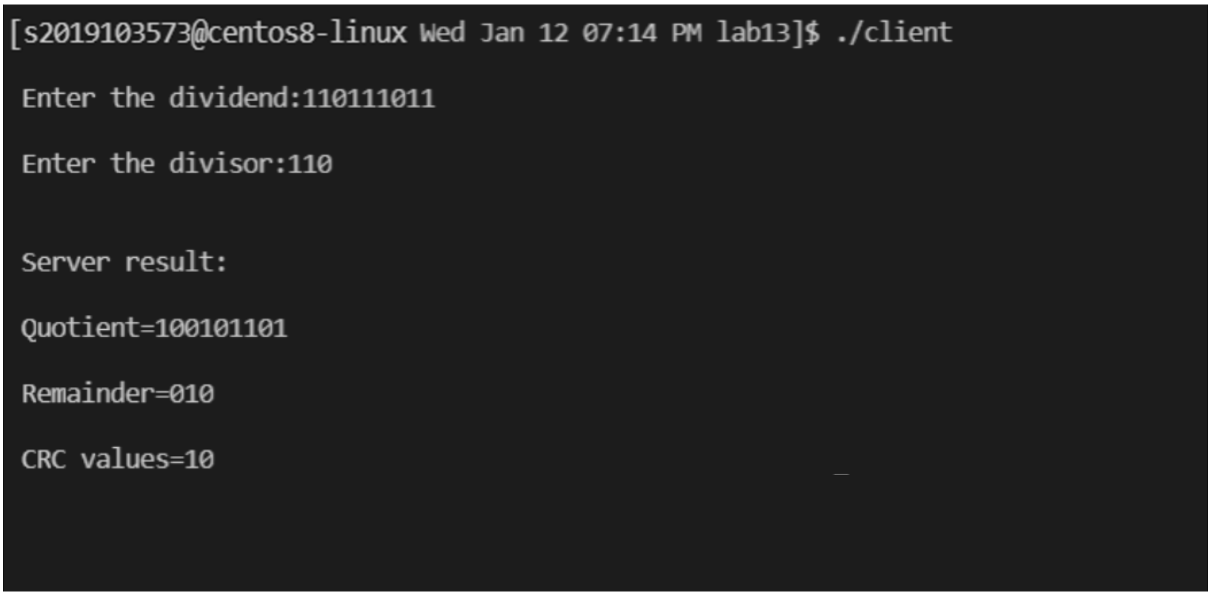
OUTPUT :

SERVER



```
[s2019103573@centos8-linux Wed Jan 12 07:14 PM lab13]$ ./server

  Binded...
  Dividend:110111011
  Divisor:110
  Dividend with zero appended:11011101100
  Quotient=100101101
  Remainder=010
  CRC values: 10
```

CLIENT



```
[s2019103573@centos8-linux Wed Jan 12 07:14 PM lab13]$ ./client

 Enter the dividend:110111011

 Enter the divisor:110


 Server result:

 Quotient=100101101

 Remainder=010

 CRC values=10
```