Q. Write a program to generate CRC code for the dividend.

```
#include<stdio.h>
void main()
{
    int dividend[11], divison[4], Q[8];
    int i,j;
    printf("Enter 4 bit for divison: ");
    for (i=0; i<4; i++)
        scanf("%d", &divison[i]);
    printf("\nEnter 8 bit for dividend:");
    for(i=0;i<8;i++)
        scanf("%d",&dividend[i]);
    for(i=8;i<11;i++)
        dividend[i]=0;
    printf("Dividend:");
    for(i=0;i<11;i++)
        printf("%d",dividend[i]);
    printf("\nDivison:");
    for(i=0;i<4;i++)
        printf("%d",divison[i]);
    for(i=0;i<8;i++)
    {
        if(dividend[i]==1)
        {
            Q[i]=1;
            for(j=i;j<=(i+3);j++)
                dividend[j]=dividend[j] ^ divison[j-i];
        }
        else
            Q[i]=0;
    }
    printf("\nDisplay quotient: ");
    for(i=0;i<8;i++)
        printf("%d",Q[i]);
        printf("\nDisplay remainder : ");
        for(i=8;i<11;i++)
            printf("%d",dividend[i]);
}
```

Output :-

```
Enter 4 bit for divisor : 1 0 0 1
Enter 8 bit for dividend : 1 0 0 1 1 0 0 1 0 0 0
Dividend : 10011001000
Divisor : 1001
Display quotient : 10001000
Display remainder : 000
```

Q:- CHECKSUM :-

```
#include<stdio.h>
void main()
{
    int i,j,m,n,temp;
    printf("Enter number of blocks (m) : ");
    scanf("%d",&m);
    printf("Enter number of bits (n) : ");
    scanf("%d",&n);
    int B[m][n], sum[n], carry=0;
    int checksum[n];
    for(i=0;i<n;i++)
        sum[i]=0;
    printf("\n");
    for(i=0;i<m;i++)
    {
        printf("Enter %d bits of B[%d] block : ",n,i);
        for(j=0;j<n;j++)
            scanf("%d",&B[i][j]);
    }
for(i=0;i<n;i++)
{  for(j=n-1; j>=0;j--)
   {
     temp=carry;
     carry=(carry&B[i][j]) | (B[i][j] & sum[j]) |
                         (sum[j] & carry)
      sum[j]=sum[j]^B[i][j]^temp;
   }
}
while(carry!=0)
{
    for(t=n-1; i>=0; i--)
    {
        if(carry==0 && sum[i]==0)
        {
            sum[i]=0;
            carry=0;
        }
        else if(carry==1 &&sum[i]==1)
        {
            sum[i]=0;
            carry=1;
        }
        else
        {
            sum[i]=1;
            carry=0;
        }
    }
}
printf(" SUM = \t\t");
for(i=0; i<n; i++)
      printf("%d\t", sum[i]);
for(i=0;i<n;i++)
      checksum[i]=!sum[i];
printf("\nCHEKSUM : \t");
for(i=0;i<n;i++)
     printf("%d\t",checksum[i]);  }
```

Scanned with CamScanner

Output :-

```
Enter number of blocks(m): 3
Enter number of bits(n): 4
Enter 4 bits of B[0] block: 1 1 0 1
Enter 4 bits of B[1] block: 1 1 1 0
Enter 4 bits of B[2] block: 0 1 0 0
SUM =          0   0   0   1
CHECKSUM = 1   1   1   0
```

## Q Stop and Wait protocol :-

```
#include<stdio.h>
#include<time.h>
clock_t sender(int [], int);
void error(int []);
void receiver();
void resend(int, int, float, clock_t);
float delay;
int copy[4], i=-1;
int main()
{
    int F[4],n,i;
    int seq_no;
    printf("Enter the number of frames:");
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        seq_no=i%2;
        clock_t start =sender(F,seq_no);
        error(F);
        receiver(F,start,seq_no);
    }
    return 0;
}
```

```
clock_t  sender(int F[], int seq_no)
{
    int i;
    int c=0;
    ++j;
    for( i=0; i<=2; i++)
    {
        printf("Enter F%d[%d]=  ",j,i);
        scanf("%d",&F[i]);
        copy[i]=F[i];
        if(F[i]==1)
            c++;
    }
    if(c%2==0)
        F[3]=copy[3]=0;
    else
        F[3]=copy[3]=1;
    printf("Parity bit is =%d\n",F[3]);
    printf("Enter delay value:");
    scanf("%f",&delay);
    delay=delay*(float)CLOCKS_PER_SEC;
    clock_t start=clock();
    while((clock()-start)<delay);
    return start;
}
void error(int F[])
{
    int i;
    char ch;
    for(i=0;i<4;i++)
    {
        printf("Are you want to add error in %dth bit??:",i);
        getchar();
        ch=getchar();
        if(ch=='y' ||ch=='Y')
        {
            printf("Enter a bit to create error:");
            scanf("%d",&F[i]);
        }
    }
}
void receiver(int F[],clock_t start,int seq_no)
{
    int c=0;
    int i,ack_no=0;
    for(i=0;i<4;i++)
    {   if(F[i]==1)
            c++;
    }
    if(c%2==0)
    {
        ack_no=(seq_no+1)%2;
        printf("\nSend ACK number=%d\n",ack_no);
        printf("%dth data sent successfully.\n\n",seq_no);
    }
    else
    {
        printf("\nData lost OR ACK lost OR Data damaged");
        resend(ack_no;seq_no,delay, start);
    }
}
void resend(int ack_no,int seq_no,float delay,clock_t start)
{   int i;
    if(((clock()-start)>=delay)||(ack_no!=(seq_no+1)%2))
    {
        printf("\nResend data is: ");
        for(i=0;i<4;i++)
            printf("%d",copy[i]);
        ack_no=(seq_no+1)%2;
        printf("\nSend ACK number=%d\n",ack_no);
        printf("\n%dth data sent successfully.",seq_no);
    }
}
```

Output:-

```
Enter the number of frames: 2
Enter F0[0]=1
Enter F0[1]=1
Enter F0[2]=1

Parity bit is =1

Enter delay value=2
Are you want to add error in 0th bit??: y
Enter a bit to create error: 0
Are you want to add error in 1th bit??: y
Enter a bit to create error: 0
Are you want to add error in 2th bith?? n
Are you want to add error in 3th bit?? n

Send ACK number=1
0th data sent successfully.

Enter F1[0]=0
Enter F1[1]=0
Enter F1[2]=1

Parity bit is = 1
```

```
Enter delay value=3
Are you want to add error in 0th bit?? y
Enter a bit to create error;1
Are you want to add error in 1th bit?? n
Are you want to add error in 2th bit?? n
Are you want to add error in 3th bit?? n

Data lost OR ACK lost OR Data damaged

Resend data is ; 0011

Send ACK number = 0
1th data sent successfully.
```