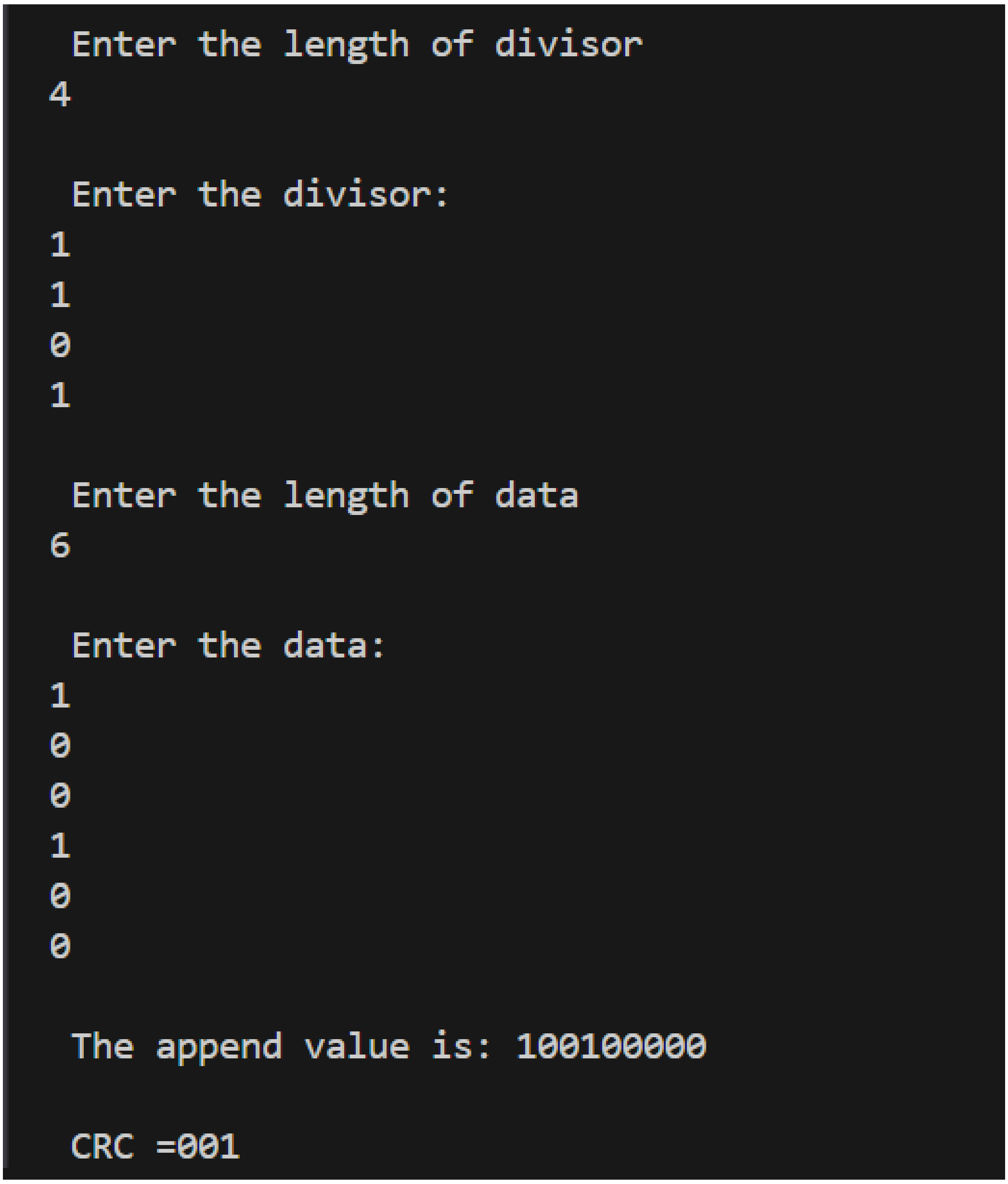
‘Cyclic Redundancy Check’ method to encode data at sender side and decoding the data at receiver end for error detection.

Sender side :

```
#include<stdio.h>

int main(){
   int m[20],p[20],d[10],i,j,k,len,rem[10],divlen;

   printf("\n Enter the length of divisor\n");
   scanf("%d",&divlen);
   printf("\n Enter the divisor:\n");
   for(i=0;i<divlen;i++)
   {
    scanf("%d",&d[i]);
   }
   printf("\n Enter the length of data\n");
   scanf("%d",&len);
   printf("\n Enter the data:\n");
   for(i=0;i<len;i++)
   {
    scanf("%d",&m[i]);
    p[i]=m[i];
   }
   for(i=len;i<len+(divlen-1);i++)
   {
    m[i]=0;
   }
   printf("\n The append value is: ");
```

```
    for(i=0;i<len+(divlen-1);i++)
    {
     printf("%d",m[i]);
    }

// ***********X-OR operation*************//

    for(i=0;i<len;i++)
      {
     if(m[i]==1)
       {
      for(j=0;j<divlen;j++)
        {
       if(m[i+j]==d[j])
        rem[j]=0;
       else
        rem[j]=1;
        }
    for(k=0;k<divlen;k++)
    {
     m[i+k]=rem[k];
    }
    }
    else
    {
      for(k=0;k<divlen;k++)
      {
       rem[k]=m[i+k];
      }
    }
    }
    printf("\n");
    printf("\n CRC =");
    for(i=1;i<divlen;i++)
    {
     printf("%d",rem[i]);
    }
    return 0;
}
```

Output :



```
 Enter the length of divisor
4

 Enter the divisor:
1
1
0
1

 Enter the length of data
6

 Enter the data:
1
0
0
1
0
0

 The append value is: 100100000

 CRC =001
```

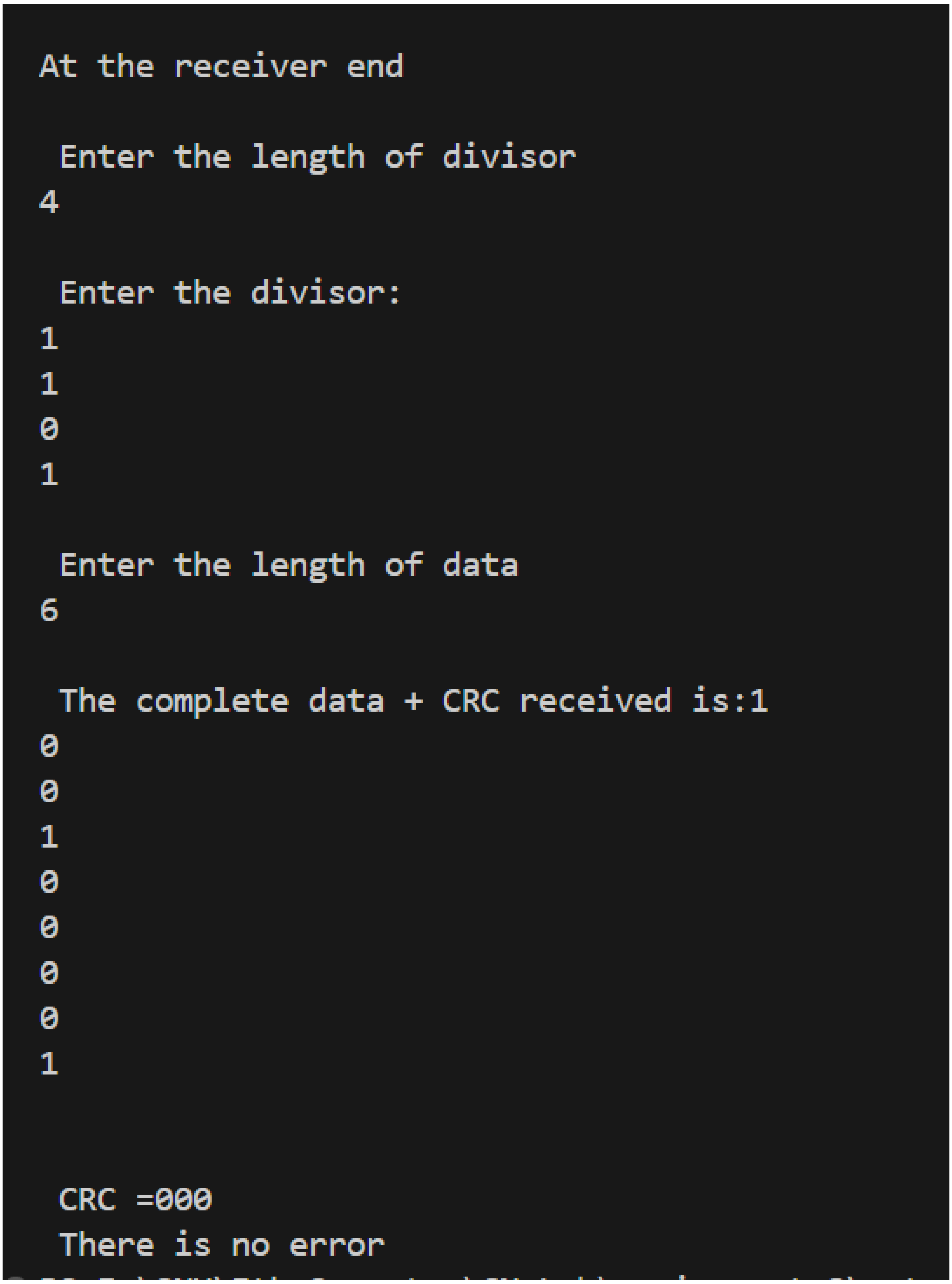
Receiver Side :

```
#include<stdio.h>

int main(){
    int i,j,k,divlen,len,m1[20],d[10],rem1[10];
    printf("\n\rAt the receiver end\n");
    printf("\n Enter the length of divisor\n");
    scanf("%d",&divlen);
    printf("\n Enter the divisor:\n");
    for(i=0;i<divlen;i++)
    {
     scanf("%d",&d[i]);
    }
    printf("\n Enter the length of data\n");
    scanf("%d",&len);

    printf("\n The complete data + CRC received is:");
    for(i=0;i<len+(divlen-1);i++)
    {
     scanf("%d",&m1[i]);
    }

      for(i=0;i<len;i++)
    {
     if(m1[i]==1)
     {
      for(j=0;j<divlen;j++)
      {
       if(m1[i+j]==d[j])
        rem1[j]=0;
```

```
   else
    rem1[j]=1;
  }
for(k=0;k<divlen;k++)
{
 m1[i+k]=rem1[k];
}
}
else
{
  for(k=0;k<divlen;k++)
  {
   rem1[k]=m1[i+k];
  }
}
}

printf("\n");
printf("\n CRC =");
j=0;
for(i=1;i<divlen;i++)
{
 printf("%d",rem1[i]);
 j=j+rem1[i];
}

if(j==0)
  printf("\n There is no error\n");
else
  printf("\n There is error\n");
```

```
    return 0;
}
```

Output :



```
At the receiver end

 Enter the length of divisor
4

 Enter the divisor:
1
1
0
1

 Enter the length of data
6

 The complete data + CRC received is:1
0
0
1
0
0
0
0
1


 CRC =000
 There is no error
```