Error detecting code using CRC-CCITT (16 bits)

SHALINI.P
1BM19CS406.

```
#include<string.h>
#include<stdio.h>
#define N strlen(gen)
char modif[20], checksum[20], gen[20];
int a,e,c,b;
 Void xor()
 {
   for(c=1; c<N; c++)
   checksum[c] = ((checksum[c] = gen[c])?'0':'1');
 }
 void crc()
 {
   for(e=0;e<N;e++)
   checksum[e] =modif[e];
   do
   {
     if(checksum[0] =='1')
     xor();
     for(c=0;c<N-1;c++)
     checksum[c] = checksum[c+1];
     checksum[c] = modif[e++];
   }
   while(e<=a+N-1);
 }
 int main()
 {
   int flag=0;
   strcpy(gen,"10001000000100001");
   printf("enter data");
   scanf("%s",modif);
   printf("\n-----\n");
   printf("\n generating polynomial: %s",gen);
   a= strlen(modif);
   for(e=a;e<a+N-1;e++)
   modif[e]='0';
```

```
crc();
printf("checksum is : %s", checksum);
for(e=a; e<a+N-1; e++)
modif[e] = checksum[e-a];
printf("\n final codeword is : %s", modif);
printf("\n ----\n");
printf("\n test error detection 0(yes) 1(no)? :");
scanf("%d", &e);
if(e==0)
{
    do
    {
        printf("Enter position where error is to be inverted");
        scanf("%d", &e);
    } while(e==0 || e>a+N-1);
    modif[e-1] = (modif[e-1]=='0')?'1':'0';
    printf("erroneous data : %s\n", modif);
}
```