Cyclic redundancy check implementation using C:

Program:

```
#include<stdio.h>
#include<string.h>
#define N strlen(g)

char t[28],cs[28],g[]="10001000000100001";
int a,e,c;

void xor(){
    for(c = 1;c < N; c++)
    cs[c] = (( cs[c] == g[c])?'0':'1');
}

void crc(){
    for(e=0;e<N;e++)
        cs[e]=t[e];
    do{
        if(cs[0]=='1')
            xor();
        for(c=0;c<N-1;c++)
            cs[c]=cs[c+1];
        cs[c]=t[e++];
    }while(e<=a+N-1);
}

int main()
{
    printf("\nEnter data : ");
    scanf("%s",t);
    printf("\n---------------------------------------");
    printf("\nGeneratng polynomial : %s",g);
    a=strlen(t);
    for(e=a;e<a+N-1;e++)
        t[e]='0';
    printf("\n---------------------------------------");
    printf("\nModified data is : %s",t);
    printf("\n---------------------------------------");
    crc();
    printf("\nChecksum is : %s",cs);
    for(e=a;e<a+N-1;e++)
        t[e]=cs[e-a];
    printf("\n---------------------------------------");
```

```
    printf("\nFinal codeword is : %s",t);
    printf("\n--------------------------------------");
    printf("\nTest error detection 0(yes) 1(no)? : ");
    scanf("%d",&e);
    if(e==0)
    {
        do{
            printf("\nEnter the position where error is to be
inserted : ");
            scanf("%d",&e);
        }while(e==0 || e>a+N-1);
        t[e-1]=(t[e-1]=='0')?'1':'0';
        printf("--------------------------------------");
        printf("\nErroneous data : %s\n",t);
    }
    crc();
    for(e=0;(e<N-1) && (cs[e]!='1');e++);
        if(e<N-1)
            printf("\nError detected\n");
        else
            printf("\nNo error detected\n\n")
     return 0;
}
```

Output:

```
Enter data : 1101

---------------------------------------
Generatng polynomial : 10001000000100001
---------------------------------------
Modified data is : 11010000000000000000
---------------------------------------
Checksum is : 1101000110101101
---------------------------------------
Final codeword is : 11011101000110101101
---------------------------------------
Test error detection 0(yes) 1(no)? : 0

Enter the position where error is to be inserted : 2
---------------------------------------
Erroneous data : 10011101000110101101
Error detected
```