```
#include<stdio.h>
#include<string.h>
#define N strlen(g)
//declare the header libraries
char t[50], cs[50], g[50];
int a,e,c;


void xor()
{
	for(c=1;c<N;c++)
	//
		cs[c]=((cs[c]==g[c])?'0':'1');
		//Checking the XOR operation. If both operands are same, then output will b "0"
otherwise its "1".
}


void crc()
{
	for(e=0;e<N;e++)
	//Consider only first FIVE bits from the modified data
		cs[e]=t[e];
		//Copy those first FIVE bits to CHECKSUM cs[e] from t[e]
	do{
		if(cs[0]=='1')
		//If first leftmost bit is 1 then perform XOR operation
			xor();
			//Calling XOR function
		for(c=0;c<N-1;c++)
		//Performing XOR operation at the first iteration for FIVE bits (0 to N-1)
			cs[c]=cs[c+1];
			//Peform the same for all the data by right shift by 1
	cs[c]=t[e++];
	} while(e<=a+N-1);
	//Continue the operation for the entire data.
}

int main(){
	printf("\n Enter the data: ");
	//Enter the data as 1101011011
	scanf("%s", t);
	// Data stored in a string t
	printf("\n Enter the generator polynomial: ");
	scanf("%s", g);
	//Enter the generator polynomial: Since we have hard coded the GP as 10011
	a=strlen(t);
	// "a" defines the total length of the data
	for(e=a;e<a+N-1;e++)
	//Appending N-1 zeros to the data where N is the length of the GP
		t[e]='0';
		//t[e] defines appending zeros from e=a;e<a+N-1;e++
```

```
	printf("\n Modified data is: %s", t);
	//MOdified data is 11010110110000
	crc();
	//Call CRC function
	printf("\n Checksum is: %s", cs);
	//Print the checksum after XOR operation
	for(e=a;e<a+N-1;e++)
	//To append the checksum value instead of N-1 zeros in total length of the data
		t[e]=cs[e-a];
		//The remodified data with checksum (FINAL CODEWORD)
	printf("\n The final codeword is: %s", t);
	//Print the final codeword
	printf("\n Test error detection: 0 for YES and 1 for NO: ");
	//To check for error detection
	scanf("%d", &e);
	if(e==0)
	//If the value of "e" is 0
	{
	do {
		printf("\n Enter the position where error is to be inserted: ");
		//Specify the position
		scanf("%d", &e);
		//Say for example, e=6
	} while(e==0||e>a+N-1);
	//WHILE states the boundary, means ranging for 0 to a+N-1

	t[e-1] = (t[e-1]=='0')?'1':'0';
	//Changing the bit from 0 to 1 and vice versa for error detection
	printf("\n Erroneous data: %s\n",t);
	}
	crc();
	for(e=0; (e<N-1)&&(cs[e]!='1'); e++);
	//If CHECKSUM is not equal to 1 then error is detected else no error
		if(e<N-1)
			printf("\n Error detected \n \n");
		else
			printf("\n No error detected \n \n");
return 0;
}
```