WEEK 3

AIM:

To implement Hamming code and CRC for Error Handling

PROCEDURE:

PROGRAM-1: Hamming code

```
#include<stdio.h>

void client(int dataword[12])

{

   int c1,c2,c4,c8;//c1 c2 c4 c8 are check bits

   printf("\nVerifying Dataword");


c1=dataword[0]^dataword[2]^dataword[4]^dataword[6]^dataword[
8]^datawo rd[10];


c2=dataword[1]^dataword[2]^dataword[5]^dataword[6]^dataword[
9]^datawo rd[10];


c4=dataword[3]^dataword[4]^dataword[5]^dataword[6]^dataword[
11];


c8=dataword[7]^dataword[8]^dataword[9]^dataword[10]^dataword
[11];

   if(c1==0 && c2==0 && c4==0 && c8==0)
```

```
    {
        printf("\nMessage not Corrupted");
    }
    else
    {
        printf("\nMessage Corrupted");
    }
}
void server(int *dataword)
{
    int p1,p2,p4,p8;//p1 p2 p4 p8 are parity bits

p1=dataword[2]^dataword[4]^dataword[6]^dataword[8]^dataword[
10];
    dataword[0]=p1;

p2=dataword[2]^dataword[5]^dataword[6]^dataword[9]^dataword[
10];
    dataword[1]=p2;
    p4=dataword[4]^dataword[5]^dataword[6]^dataword[11];
    dataword[3]=p4;
    p8=dataword[8]^dataword[9]^dataword[10]^dataword[11];
    dataword[7]=p8;
```

```
    for(int i=0;i<12;i++)
    {
        printf("%d->%d ",(i+1),dataword[i]);
    }
    client(dataword);
}
int main()
{
    int dataword[12],i;
    printf("Enter 8 bit Data word:\n");
    for(i=0;i<12;i++)
    {
        if(i==0 || i==1 || i==3 || i==7)
        {
            dataword[i]=0;
            continue;
        }
        scanf("%d",&dataword[i]);//input should be either 0 or 1
    }
    server(dataword);
    return 0; }
```