1. **Write a program to compute the Hamming Distance between two equal lengths binary string.**

```c
#include <stdio.h>
#include<stdlib.h>

int main() {
    int x[]={1,0,1,1,1,0,1,1};
    int y[]={1,0,0,1,1,0,0,1};
    int a[8];

     int i;

    for( i=0;i<8;i++){
        a[i]=x[i]^y[i];
    }

    int count=0;

  printf("the hamming distance codeword is ");
    for( i=0;i<8;i++){
        printf("%d",a[i]);
        if(a[i]==1)
        { count++;}
    }



    printf("\nhamming distance between x and y is %d.", count);
 return 0;
}
```

**2. Write a program to compute the minimal Hamming Distance among multiple
equal length codeword. Nd**
        Using
        dmin=2t + 1 and
        dmin = s+1
       calculate the value of 's' and 't'....

```c
#include <stdio.h>
#include<stdlib.h>
int min(int a,int b,int c,int d){

    int min = a;

    if (b < min) min = b;
    if (c < min) min = c;
    if (d < min) min = d;

    return min;

}
int main() {
    int x[]={1,0,1,1,1,0,1,1};


    int y[4][8];
    int a[8];
    int b[8];
    int c[8];
    int d[8];
     int i,j;
    for( i=0;i<4;i++){
        for( j=0;j<8;j++){
            y[i][j]=rand()%2;

    }
    }
      for( i=0;i<4;i++){
        for( j=0;j<8;j++){
            printf("%d\t",y[i][j]);
        }
        printf("\n");
    }

    for( i=0;i<8;i++){
        a[i]=y[0][i]^x[i];
        b[i]=y[1][i]^x[i];
        c[i]=y[2][i]^x[i];
        d[i]=y[3][i]^x[i];
    }


    int count1=0;
    int count2=0;
    int count3=0;
```

```c
    int count4=0;


    for( i=0;i<8;i++){

        if(a[i]==1)
        { count1++;}

        if(b[i]==1)
        {count2++;}

        if(c[i]==1)
        {count3++;}

        if(d[i]==1)
        {count4++;}

    }

  printf("\nhamming distance between x and a is %d", count1);
     printf("\nhamming distance between x and b is %d", count2);
         printf("\nhamming distance between x and c is %d", count3);
            printf("\nhamming distance between x and d is %d",count4);
 int dmin=min(count1,count2,count3,count4);
 printf("\nMinimum number: %d\n", dmin);

 int s,t;

 t=(dmin-1)/2;
 s=(dmin-1);
   printf("\nmaximum number of error can be detected is %d\n", s);
   printf("\nmaximum number of error can be corrected is %d\n", t);



    return 0;
}
```