

Experiment-3

Write a Program to implement data link layer farming method checksum.

Program:-

```
#include<stdio.h>
#include<string.h>
int main()
{
char a[20],b[20],sum[20],complement[20];
int i,length;
printf("enter first binary string:\n");
scanf("%s",&a);
printf("enter second binary string:\n");
scanf("%s",&b);
if(strlen(a)==strlen(b))
{
length=strlen(b);
char carry='0';
for(i=length-1;i>=0;i--)
{
if(a[i]=='0' &&b[i]=='0'&&carry=='0')
{
sum[i]='0';
carry='0';
}
else if(a[i]=='0' &&b[i]=='0'&&carry=='1')
{
sum[i]='1';
carry='0';
}
else if(a[i]=='0' &&b[i]=='1'&&carry=='0')
{
sum[i]='1';
carry='0';
}
else if(a[i]=='0' &&b[i]=='1'&&carry=='1')
{
sum[i]='0';
carry='1';
}
else if(a[i]=='1' &&b[i]=='0'&&carry=='0')
{
sum[i]='1';
carry='0';
}
}
```

```

else if(a[i]=='1' && b[i]=='0' && carry=='1')
{
    sum[i]='0';
    carry='1';
}
else if(a[i]=='1' && b[i]=='1' && carry=='0')
{
    sum[i]='0';
    carry='1';
}
else if(a[i]=='1' && b[i]=='1' && carry=='1')
{
    sum[i]='1';
    carry='1';
}
else
    break;
}
printf("\nSum=%c%s", carry, sum);
for(i=0; i<length; i++)
{
    if(sum[i]=='0')
        complement[i]='1';
    else
        complement[i]='0';
}
if(carry=='1')
    carry='0';
else
    carry='1';
printf("\nChecksum=%c%s\n", carry, complement);
}
else
    printf("\nWrong input String");
}

```

Output:-

```

[20A91A0568@Linux ~]$ vi exp3.c
[20A91A0568@Linux ~]$ cc exp3.c
[20A91A0568@Linux ~]$ ./a.out
enter first binary string:
1010110
enter second binary string:
0101001

Sum=01111111
Checksum=10000000

```

Experiment-4

Write a program for Hamming Code generation for error detection and correction.

Program:

```
#include<stdio.h>
void main() {
    int data[10];
    int dataatrec[10],c,c1,c2,c3,i;
    printf("Enter 4 bits of data one by one\n");
    scanf("%d",&data[0]);
    scanf("%d",&data[1]);
    scanf("%d",&data[2]);
    scanf("%d",&data[4]);

    //Calculation of even parity
    data[6]=data[0]^data[2]^data[4];
    data[5]=data[0]^data[1]^data[4];
    data[3]=data[0]^data[1]^data[2];

    printf("\nEncoded data is\n");
    for(i=0;i<7;i++)
        printf("%d",data[i]);

    printf("\n\nEnter received data bits one by one\n");
    for(i=0;i<7;i++)
        scanf("%d",&dataatrec[i]);

    c1=dataatrec[6]^dataatrec[4]^dataatrec[2]^dataatrec[0];
    c2=dataatrec[5]^dataatrec[4]^dataatrec[1]^dataatrec[0];
    c3=dataatrec[3]^dataatrec[2]^dataatrec[1]^dataatrec[0];
    c=c3*4+c2*2+c1 ;

    if(c==0) {
        printf("\nNo error while transmission of data\n");
    }
    else {
        printf("\nError on position %d",c);
    }

    printf("\nData sent : ");
```

```

        for(i=0;i<7;i++)
            printf("%d",data[i]);

printf("\nData received : ");
        for(i=0;i<7;i++)
            printf("%d",dataatrec[i]);
printf("\nCorrect message is\n");

//if erroneous bit is 0 we complement it else vice versa

if(dataatrec[7-c]==0)
if(dataatrec[7-c]==0)
dataatrec[7-c]=1;
    else
dataatrec[7-c]=0;
for (i=0;i<7;i++) {
printf("%d",dataatrec[i]);
}
}
}

```

Output:

```

[20A91A0568@Linux ~]$ vi exp4.c
[20A91A0568@Linux ~]$ cc exp4.c
[20A91A0568@Linux ~]$ ./a.out
Enter 4 bits of data one by one
1
0
1
0

Encoded data is
1010010

Enter received data bits one by one
1
0
1
0
0
1
0

No error while transmission of data

```