**Experiment No : 1**

**Aim :** To Write a code in C/C++ to implement Hamming Code in Computer Networking.

**C++ Program to Implement Hamming Code**

```cpp
#include<iostream>
#include<conio.h>
using namespace std;
int main()
{
    int data[10];
    int dataatrec[10],c,c1,c2,c3,i;
    cout<<"Enter 4 bits of data to be transferred one by one: \n";
    cin>>data[0];
    cin>>data[1];
    cin>>data[2];
    cin>>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];

    cout<<"\nEncoded data is: \n";
    for(i=0;i<7;i++)
    cout<<data[i];

    cout<<"\n\nEnter received data bits one by one: \n";
    for(i=0;i<7;i++)
    cin>>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) {
            cout<<"\nNo error while transmission of data: \n";
    }
        else {
            cout<<"\nError on position: "<<c;

            cout<<"\nData sent : ";
            for(i=0;i<7;i++)
    cout<<data[i];

            cout<<"\nData received : ";
    for(i=0;i<7;i++)
        cout<<dataatrec[i];

            cout<<"\nCorrect message is: \n";
```

```cpp
                //if errorneous bit is 0 we complement it else vice versa
                if(dataatrec[7-c]==0)
                        dataatrec[7-c]=1;
        else
                        dataatrec[7-c]=0;
                for (i=0;i<7;i++) {
                        cout<<dataatrec[i];
                }
        }

        return 0;
}
```

**Output :**

```
■ "C:\Users\adity\OneDrive\Desktop\cg expts\test\hamming.exe"
Enter 4 bits of data to be transferred one by one:
1
0 1 1

Encoded data is:
1010101

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

Error on position: 1
Data sent : 1010101
Data received : 1010100
Correct message is:
1010101
Process returned 0 (0x0)    execution time : 16.282 s
Press any key to continue.
```

```
■ "C:\Users\adity\OneDrive\Desktop\cg expts\test\hamming.exe"
Enter 4 bits of data to be transferred one by one:
1 0 0 1

Encoded data is:
1001100

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

No error while transmission of data:

Process returned 0 (0x0)    execution time : 15.987 s
Press any key to continue.
```

**Conclusion :** The cpp code for implementing Hamming Code concept was written and executed successfully with o/p.

**Experiment Number : 2**

**Aim :** To Write a code in C/C++ to implement Cyclic Redundancy Check (CRC) concept in Computer Networking.

**C++ Program to Implement Cyclic Redundancy Code**

```cpp
#include<iostream>
using namespace std;
void division(int temp[],int gen[],int n,int r)

{

    for(int i=0;i<n;i++)

    {

        if (gen[0]==temp[i])

        {

            for(int j=0,k=i;j<r+1;j++,k++)

            if(!(temp[k]^gen[j]))

                temp[k]=0;

            else

                temp[k]=1;}}

}

int main()
{
    int n,r,message[50],gen[50],temp[50];
    cout<<"At Sender's End "<<endl;
    cout<<"Enter the number of message bits : ";
    cin>>n;
    cout<<"Enter the number of generator bits : ";
    cin>>r;

    cout<<"Enter the message : ";
    for(int i=0;i<n;i++)
        cin>>message[i];
            cout<<"Enter the generator : ";
            for(int i=0;i<r;i++)
        cin>>gen[i];
                    r--;
            for(int i=0;i<r;i++)
        message[n+i] = 0;
            for(int i=0;i<n+r;i++)
        temp[i] = message[i];
            division(temp,gen,n,r);
            cout<<"CRC : ";
            for(int i=0;i<r;i++)
            {
        cout<<temp[n+i]<<" ";
            message[n+i] = temp[n+i];
    }
```

```cpp
    cout<<endl<<"Transmitted Message : ";
        for(int i=0;i<n+r;i++)
    cout<<message[i]<<" ";
        cout<<endl<<endl<<"At Receiver's End "<<endl;
        cout<<"Enter the received message : ";
        for(int i=0;i<n+r;i++)
    cin>>message[i];
        for(int i=0;i<n+r;i++)
    temp[i] = message[i];
        division(temp,gen,n,r);
        for(int i=0;i<r;i++)
        {
    if(temp[n+i])
    {
        cout<<"Error detected in received message.";
        return 0;
    }}
    cout<<endl<<"No error in the Message";
        cout<<"\nReceived Message/Original Message : ";
        for(int i=0;i<n;i++)
        cout<<message[i]<<" ";
        return 0;
}
```

**Output :**

```
■ "C:\Users\adity\OneDrive\Desktop\cg expts\test\crc.exe"

At Sender's End
Enter the number of message bits : 6
Enter the number of generator bits : 3
Enter the message : 1 0 0 0 1 1
Enter the generator : 1 0 1
CRC : 0 1
Transmitted Message : 1 0 0 0 1 1 0 1

At Receiver's End
Enter the received message : 1 0 0 0 1 1 0 1

No error in the Message
Received Message/Original Message : 1 0 0 0 1 1
Process returned 0 (0x0)    execution time : 21.504 s
Press any key to continue.
```

```
■ "C:\Users\adity\OneDrive\Desktop\cg expts\test\crc.exe"

At Sender's End
Enter the number of message bits : 7
Enter the number of generator bits : 3
Enter the message : 1 0 0 0 1 1 1
Enter the generator : 1 0 1
CRC : 1 1
Transmitted Message : 1 0 0 0 1 1 1 1 1

At Receiver's End
Enter the received message : 1 0 0 0 0 1 1 1 1
Error detected in received message.
Process returned 0 (0x0)    execution time : 29.643 s
Press any key to continue.
```

**Conclusion :** The cpp code for implementing CRC concept was written and executed successfully with o/p.

**Deepraj Bhosale**
**181105016**