Assignment No. 3

CRC check:

```
#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];
 }
 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<<"No error in received Message.\nReceived Message : ";
 for(int i=0;i<n;i++)
    cout<<message[i]<<" ";
 return 0;
}
```

Output:

```
At Sender's End
Enter the number of message bits : 8
Enter the number of generator bits : 4
Enter the message : 1
1
0
0
1
0
0
1
Enter the generator : 1
0
0
1
CRC : 0 1 1
Transmitted Message : 1 1 0 0 1 0 0 1 0 1 1

At Receiver's End
Enter the received message : 1
1
0
0
1
0
0
1
0
1
1
No error in received Message.
Received Message : 1 1 0 0 1 0 0 1

sllab@sllab-Vostro-460:~/Desktop$ ./a.out
At Sender's End
Enter the number of message bits : 8
Enter the number of generator bits : 4
Enter the message : 1
0
```

```
1
0
1
0
1
0
Enter the generator : 1
0
1
0
CRC : 0 0 0
Transmitted Message : 1 0 1 0 1 0 1 0 0 0 0

At Receiver's End
Enter the received message : 1
1
0
1
1
0
1
0
1
0
1
Error detected in received message.
```

Hamming Code check:

```
#include<iostream>
#include<cmath>
#include<string>
using namespace std;
 class Hamming{
 string message;
 int codeword[50],temp[50];
 int n,check;
 char parity;
public:
 Hamming() {
     parity = 'E';
     message = "";
     n=check=0;
     for(int i=0;i<50;i++)     {
         temp[i]=codeword[i]=0;
     }
 }
 void generate(){
     do {
         cout<<"Enter the message in binary : ";
```

```
        cin>>message;
    }while(message.find_first_not_of("01") != string::npos);
    n=message.size();
    cout<<"Odd(O)/Even(E) Parity ? ";
    cin>>parity;
    for(unsigned int i=0;i<message.size();i++) {
        if(message[i] == '1')
            temp[i+1]=1;
        else
            temp[i+1]=0;
    }
    computeCode();
}
 void computeCode(){
    check = findr();
    cout<<"Number of Check Bits : "<<check<<endl;
    cout<<"Number of Bits in Codeword : "<<n+check<<endl;
    for(int i=(n+check),j=n;i>0;i--) {
        if((i & (i - 1)) != 0)
            codeword[i] = temp[j--];
        else
            codeword[i] = setParity(i);
    }
    cout<<"Parity Bits - ";
    for(int i=0;i<check;i++)
     cout<<"P"<<pow(2,i)<<" : "<<codeword[(int)pow(2,i)]<<"\t";
    cout<<endl;
    cout<<"Codeword :"<<endl;
    for(int i=1;i<=(n+check);i++)
        cout<<codeword[i]<<" ";
    cout<<endl;
}
 int findr() {
    for(int i=1;;i++)     {
        if(n+i+1 <= pow(2,i))
            return i;
    }
}
 int setParity(int x) {
    bool flag = true;
    int bit;
    if(x == 1)     {
        bit = codeword[x+2];
        for(int j=x+3;j<=(n+check);j++)         {
            if(j%2)             {
                bit ^= codeword[j];
            }
        }
    }
    else     {
        bit = codeword[x+1];
        for(int i=x;i<=(n+check);i++) {
```

```
            if(flag)
            {
               if(i==x || i==x+1)
                  bit = codeword[x+1];
               else
                  bit ^= codeword[i];
            }
            if((i+1)%x == 0)
               flag = !flag;
         }
      }
      if(parity == 'O' || parity == 'o')
         return !bit;
      else
         return bit;
   }
       void correct() {
      do {
         cout<<"Enter the received codeword : ";
         cin>>message;
      }while(message.find_first_not_of("01") != string::npos);
      for(unsigned int i=0;i<message.size();i++) {
         if(message[i] == '1')
            codeword[i+1]=1;
         else
            codeword[i+1]=0;
      }
      detect();
   }
   void detect() {
      int position = 0;
      cout<<"Parity Bits - ";
      for(int i=0;i<check;i++)   {
         bool flag = true;
         int x = pow(2,i);
         int bit = codeword[x];
         if(x == 1) {
            for(int j=x+1;j<=(n+check);j++) {
               if(j%2){
                  bit ^= codeword[j];
               }
            }
         }
         else {
            for(int k=x+1;k<=(n+check);k++){
               if(flag){
                  bit ^= codeword[k];
               }
               if((k+1)%x == 0)
                  flag = !flag;
            }
         }
```

```
        cout<<"P"<<x<<": "<<bit<<"\t";
        if((parity=='E' || parity == 'e') && bit==1)
            position += x;
        if((parity=='O' || parity == 'o') && bit==0)
            position += x;
    }
    cout<<endl<<"Received Codeword :"<<endl;
    for(int i=1;i<=(n+check);i++)
        cout<<codeword[i]<<" ";
    cout<<endl;
    if(position != 0)
    {
        cout<<"Error at bit : "<<position<<endl;
        codeword[position] = !codeword[position];
        cout<<"Corrected Codeword : "<<endl;
        for(int i=1;i<=(n+check);i++)
            cout<<codeword[i]<<" ";
        cout<<endl;
    }
    else
        cout<<"No Error in Received code."<<endl;
    cout<<"Received Message is : ";
    for(int i=1;i<=(n+check);i++)
        if((i & (i - 1)) != 0)
            cout<<codeword[i]<<" ";
    cout<<endl;
 }
};
 int main()
{
 char choice;
 do
 {
    Hamming a;
    cout<<"At Sender's side : "<<endl;
    a.generate();
    cout<<endl<<"At Receiver's Side : "<<endl;
    a.correct();
    cout<<endl<<"Enter another code ? (Y/N) : ";
    cin>>choice;
    cout<<endl;
 }while(choice == 'y' || choice == 'Y');
 return 0;
}
```

Output:

```
sllab@sllab-Vostro-460:~/Desktop$ ./a.out
At Sender's side :
Enter the message in binary : 1001101
Odd(O)/Even(E) Parity ? E
Number of Check Bits : 4
Number of Bits in Codeword : 11
```

Parity Bits - P1 : 0 P2 : 1 P4 : 1 P8 : 0
Codeword :
0 1 1 1 0 0 1 0 1 0 1

At Receiver's Side :
Enter the received codeword : 01110010101
Parity Bits - P1: 0 P2: 0 P4: 0 P8: 0
Received Codeword :
0 1 1 1 0 0 1 0 1 0 1
No Error in Received code.
Received Message is : 1 0 0 1 1 0 1

Enter another code ? (Y/N) : y

At Sender's side :
Enter the message in binary : 110011001100
Odd(O)/Even(E) Parity ? O
Number of Check Bits : 5
Number of Bits in Codeword : 17
Parity Bits - P1 : 1 P2 : 0 P4 : 0 P8 : 1 P16 : 1
Codeword :
1 0 1 0 1 0 0 1 1 1 0 0 1 1 0 1 0

At Receiver's Side :
Enter the received codeword : 110011001100
Parity Bits - P1: 0 P2: 0 P4: 0 P8: 0 P16: 1
Received Codeword :
1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 1 0
Error at bit : 15
Corrected Codeword :
1 1 0 0 1 1 0 0 1 1 0 0 1 1 1 1 0
Received Message is : 0 1 1 0 1 1 0 0 1 1 1 0

Enter another code ? (Y/N) : N