Experiment No: - 3

# Aim: To study the principles of channel encoding and Hamming code.

# Theory:

A major feature of digital data transmission is the myriad techniques used to protect data or speech through coding. Coding adds additional bits to the original payload to provide a means of protecting the original information. This gives the data more security, since it is possible to identify and even correct (within certain limits) data corrupted in the transmission path.

The purpose of channel coding theory is to find codes which transmit quickly, contain many valid [code words](https://en.wikipedia.org/wiki/Code_word) and can correct or at least [detect](https://en.wikipedia.org/wiki/Error_detection) many errors. While not mutually exclusive, performance in these areas is a trade off. So, different codes are optimal for different applications. The needed properties of this code mainly depend on the probability of errors happening during transmission.

## Hamming code:

In telecommunication, Hamming codes are a family of linear error-correcting codes and were invented by Richard Hamming in 1950. Hamming codes can ***detect and correct one-bit error***. By contrast, the simple parity code cannot correct errors, and can detect only an odd number of bits in error. Hamming codes are perfect codes, that is, they achieve the highest possible rate for codes with their block length and minimum distance of three.[1]

## History: -

Richard Hamming, the inventor of Hamming codes, worked at Bell Labs in the 1940s on the Bell Model V computer, an [electromechanical](https://en.wikipedia.org/wiki/Electromechanical) relay-based machine with cycle times in seconds. Input was fed in on punched paper tape, seven-eighths of an inch wide which had up to six holes per row. During weekdays, when errors in the relays were detected, the machine would stop and flash lights so that the operators could correct the problem. During after-hours periods and on weekends, when there were no operators, the machine simply moved on to the next job.

Hamming worked on weekends, and grew increasingly frustrated with having to restart his programs from scratch due to detected errors. In a taped interview Hamming said, "And so I said, 'Damn it, if the machine can detect an error, why can't it locate the position of the error and correct it?'". Over the next few years, he worked on the problem of error-correction, developing an increasingly powerful array of algorithms. In 1950, he published what is now known as Hamming Code, which remains in use today in applications such as [ECC memory](https://en.wikipedia.org/wiki/Dynamic_random_access_memory#Error_detection_and_correction).

# Code:

import java.io.\*;

import java.util.\*;

public class Ham {

public static void main(String []args){

String in,o;

boolean t=true,k=true;

StringBuilder b =new StringBuilder();

Scanner sc=new Scanner(System.in);

while(k){

System.out.println("Enter data in Binary:");

in=sc.next();

for(int i=0;i<in.length();i++){

if((in.charAt(i)-'0')>=2){

System.out.println("Wrong Input......");

t=false;

break;

}

}

if(!t)

break;

System.out.println("Correct input");

for(int i=0,j=0,ki=0;ki<in.length();i++){

if(i==Math.pow(2,j)-1){

b.append('0');

++j;

}

else{

b.append(in.charAt(ki++));

}

}

//System.out.println("The initial string:"+b.toString());

for(int i=0;i<=(int)(Math.log(b.length())/Math.log(2));i++){

int a=(int)Math.pow(2,i);

boolean par=true;

for(int j=a-1;j<b.length();){

for(int ke=j;ke<(j+a)&&ke<b.length();ke++){

if(b.charAt(ke)=='1')

par=!par;

}

j=j+2\*a;

}

if(!par){

b.setCharAt(a-1,'1');

}

}

o=b.toString();

System.out.println("The corresponding Hamming code is:"+o);

System.out.println("Enter receiver code:");

String re=sc.next();

StringBuilder ac,d=new StringBuilder();

for(int i=0;Math.pow(2, i)<re.length();i++){

d.append(re.charAt((int)Math.pow(2, i)-1));

}

ac=new StringBuilder();

for(int i=0;i<=(int)(Math.log(re.length())/Math.log(2));i++){

int a=(int)Math.pow(2,i);

boolean par=true;

for(int j=a-1;j<re.length();){

for(int ke=j;ke<(j+a)&&ke<re.length();ke++){

if(re.charAt(ke)=='1')

par=!par;

}

j=j+2\*a;

}

if(!par){

ac.append((int)Math.pow(2,i));

}

}

int index=0;

for(int i=0;i<ac.length();i++){

index+=(ac.charAt(i)-'0');

}

System.out.println("The index at which error occurred is:"+index);

System.out.println("The entered input is:"+re);

if(index!=0){

ac=new StringBuilder(re);

ac.setCharAt(index-1,ac.charAt(index-1)=='1'?'0':'1');

re=ac.toString();

}

System.out.println("The corrected output is:"+re);

b=new StringBuilder();

for(int i=0,j=0;i<re.length();i++){

if(i==Math.pow(2,j)-1){

j++;

}

else{

b.append(re.charAt(i));

}

}

System.out.println("The data transmitted was:"+b.toString());

System.out.print("Would you like to leave: press y/n ");

String c=sc.next();

if(c.equalsIgnoreCase("n"))

;

else if(c.equalsIgnoreCase("y")){

k=false;

System.out.print("Goodbye...");

}

}

}

}

# Output:

C:\Users\OWNER\Downloads>javac Ham.java

C:\Users\OWNER\Downloads>java Ham

Enter data in Binary:

011

Correct input

The corresponding Hamming code is:110011

Enter receiver code:

110001

The index at which error occurred is:5

The entered input is:110001

The corrected output is:110011

The data transmitted was:011

Would you like to leave: press y/n n

Goodbye...

# Conclusion:

Thus,