1. Implement hamming code method for error correction. Input binary data and convert input data into code words in transmitted message. Display received message and display corrected message in case of error.

**\*\*\*\*\*\*\*\*\*\*\*\*SENDER\*\*\*\*\*\*\*\*\*\*\*\***

Enter  Data       : 1011001

**Out Put**

Code Word :0110

Transmission data T : 101**0**110**1**1**10**

Do You Want to Introduce error(Y/N) : Y

Enter the Position : 2 (condition : position between 1 to T size)

**\*\*\*\*\*\*\*\*\*\*\*\*RECEIVER\*\*\*\*\*\*\*\*\*\*\*\***

Message received at the Receiver : 101011011**0**0

Code Word :0010   Error in 2nd position

After Correction of Data : 101011011**1**0



#include<stdio.h>

#include<string.h>

#include<math.h>

int pow1(int a,int b)

{

int r=1;

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

{

r=r\*a;

}

return r;

}

int check(char message [],int n)

{

int len=strlen(message);

int flag=0,i=n-1,count=0;

while(i<len)

{

int k=0;

while(k<n)

{

if(message[i++]=='1')

count++;

k++;

}

k=0;

while(k<n)

{

i++;

k++;

}

}

return count;

}

void main()

{

char mess[20],nmess[50],hcode[10];

int j=0,k=0,i=0;

printf("\*\*\* SENDER \*\*\*\n");

printf("Enter Data:");

scanf("%s",&mess);

int len=strlen(mess);

while(mess[i]!='\0')

{

if((j+1)==pow1(2,k))

{

nmess[j++]=' ';

nmess[j]='\0';

k++;

}

else

{

nmess[j++]=mess[i++];

nmess[j]='\0';

}

}

int l=0;

j=0;

printf("\nAfter appending Empty space for message:%s\n",nmess);

len=strlen(nmess);

while(pow1(2,l)<len)

{

int count=check(nmess,pow1(2,l));

if(count%2==0)

hcode[j++]='0';

else

hcode[j++]='1';

l++;

}

printf("\nCode Word:%s\n",hcode);

j=0,i=0,k=0;

while(nmess[j]!='\0')

{

if((j+1)==pow1(2,k))

{

nmess[j]=hcode[i++];

k++;

}

j++;

}

int pos;

char ch;

printf("\nTransmission Data:%s",nmess);

printf("\nDo you want to introduce error (y/n):");

scanf(" %c",&ch);

if(ch=='y')

{

printf("\nEnter the position:");

x:

scanf("%d",&pos);

if(pos>len)

{

goto x;

}

else

{

nmess[pos-1]=(nmess[pos-1]=='1')?'0':'1';

}

}

l=0,j=0;

printf("\*\*\* RECEIVER \*\*\*\n");

printf("Message received at the receiver:%s\n",nmess);

while(pow1(2,l)<len)

{

int count=check(nmess,pow1(2,l));

if(count%2==0)

hcode[j++]='0';

else

hcode[j++]='1';

l++;

}

len=strlen(hcode);

for (i = 0; i < len/2; i++)

{

int temp = hcode[i];

hcode[i] = hcode[len - i - 1];

hcode[len - i - 1] = temp;

}

printf("Code Word:%s\n",hcode);

int result=0;

pos=0;

for(i=len-1;i>=0;i--,pos++)

{

if(hcode[i]=='1')

result+=pow1(2,pos);

}

printf("Error in %d position\n",result);

nmess[result-1]=(nmess[result-1]=='1')?'0':'1';

printf("After Correction of Data:%s",nmess);

}