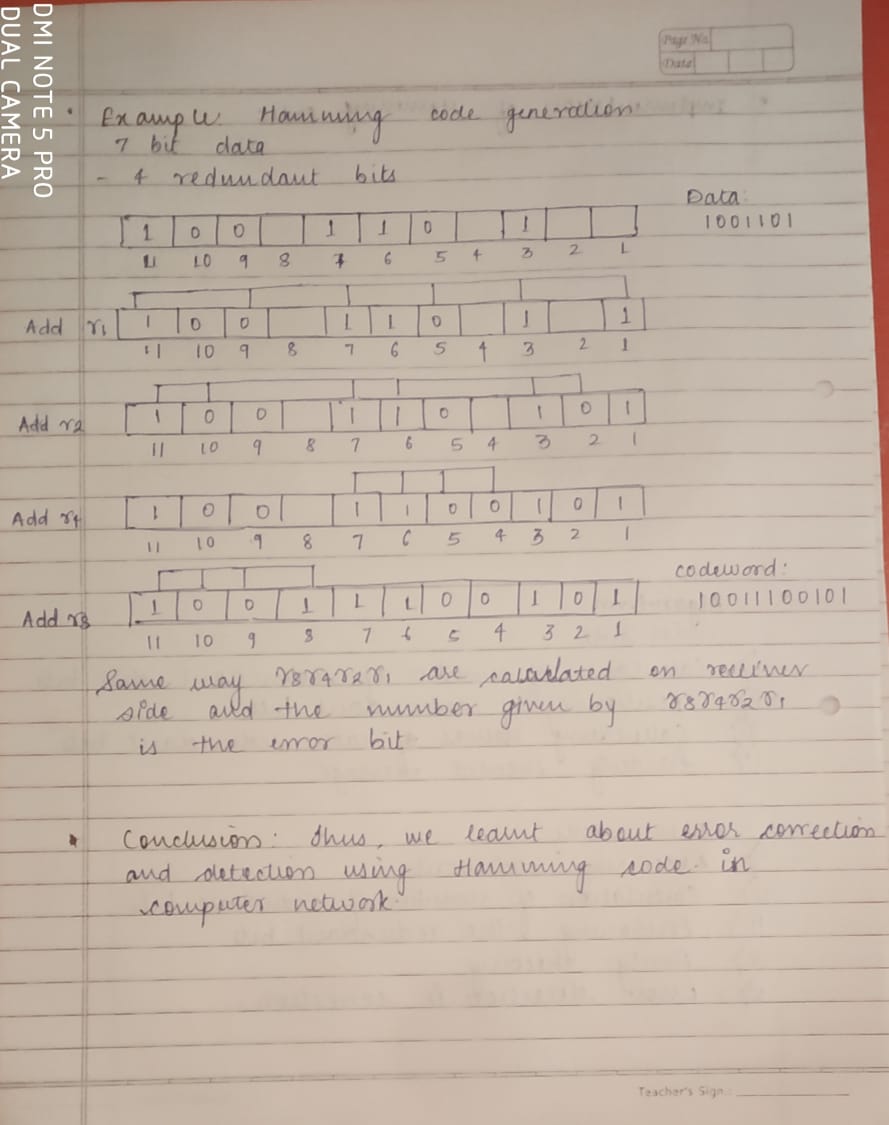
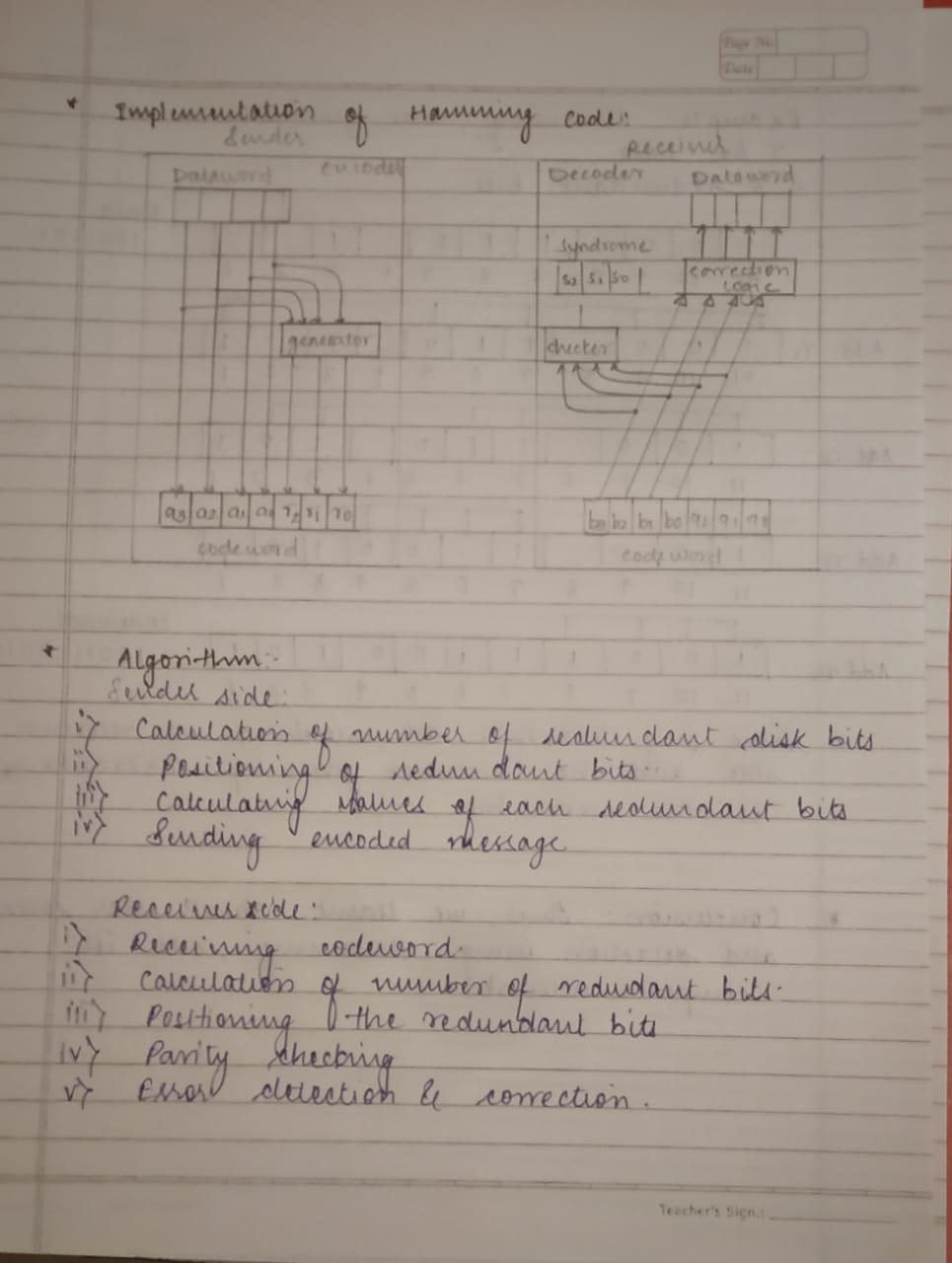
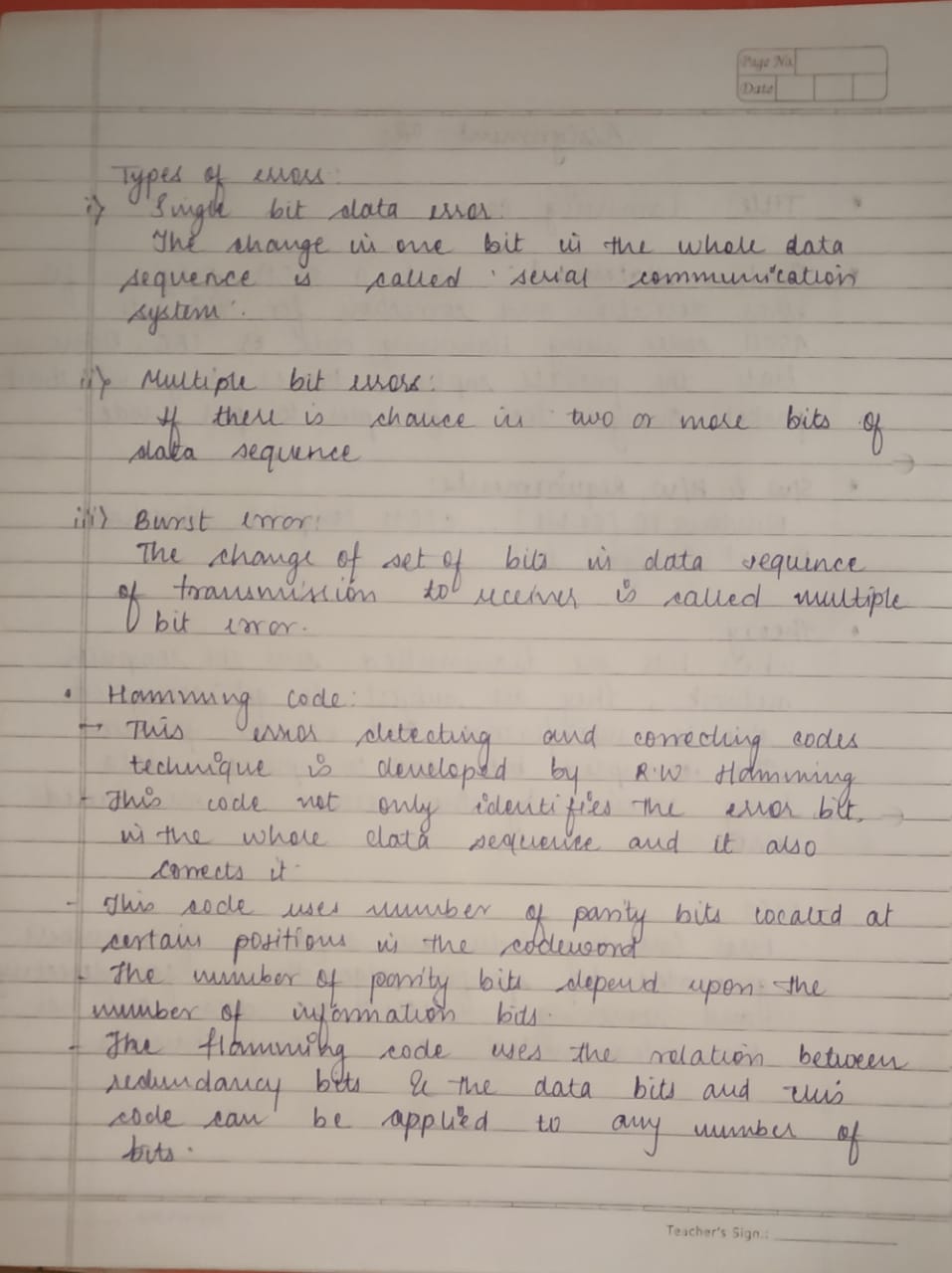
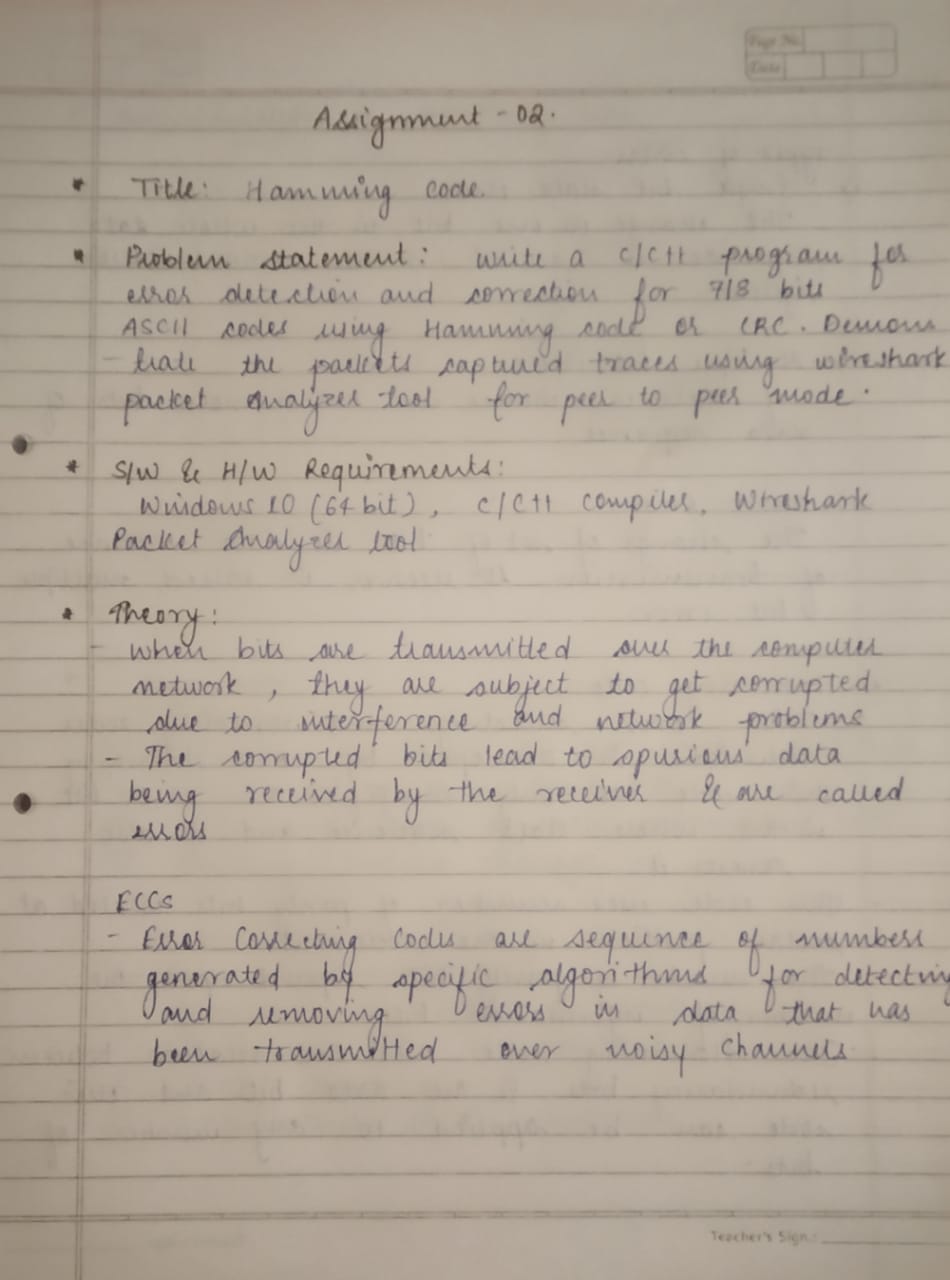
**Assignment\_2 Write Up:**

**Hamming\_Code Program:**

#include<iostream>

#include<math.h>

#include<time.h>

using namespace std;

class Hamming\_code

{

int msg[30];

int code\_sent[30];

int code\_received[30];

char parity;

int data\_bits,parity\_bits;

public:

Hamming\_code()

{

for(int i=0;i<30;i++)

{

msg[i]=code\_sent[i]=code\_received[i]=0;

}

data\_bits=parity\_bits=0;

parity='E';

}

void cal\_pbits()

{

cout<<"Enter type of Parity(E/O):";

cin>>parity;

cout<<"Enter number of data bits:";

cin>>data\_bits;

while(data\_bits+parity\_bits+1>pow(2,parity\_bits))

{

parity\_bits++;

}

cout<<"No of parity bits: "<<parity\_bits<<endl;

cout<<"Total number of data bits:"<<parity\_bits+data\_bits<<endl;

}

void read\_message()

{

cout<<"Enter Message:";

for(int i=1;i<=data\_bits;i++)

cin>>msg[i];

cout<<"Message entered is:";

for(int i=1;i<=data\_bits;i++)

cout<<msg[i];

cout<<endl;

}

void encode\_msg()

{

int d=0,p=1;

for(int i=1;i<=data\_bits+parity\_bits;i++)

{

if(i==pow(2,d))

{

code\_sent[i]=0;

d++;

}

else

{

code\_sent[i]=msg[p];

p++;

}

}

p=0;

int min,max=0,bit\_sum,k,j;

for(int i=1;i<=data\_bits+parity\_bits;i=pow(2,p))

{

p++;

bit\_sum=0;

k=i;

min=1;

max=i;

for(j=i;j<=data\_bits+parity\_bits;j=k+i)

{

for(k=j;max>=min&&k<=data\_bits+parity\_bits;++min,++k)

{

if(code\_sent[k] == 1)

bit\_sum++;

}

min=1;

}

if(parity=='E')

{

if(bit\_sum%2==0)

code\_sent[i]=0;

else

code\_sent[i]=1;

}

else

{

if(bit\_sum%2!=0)

code\_sent[i]=0;

else

code\_sent[i]=1;

}

}

}

void sent\_print()

{

cout<<"Code Sent with";

if(parity=='E')

cout<<" Even Parity:";

else

cout<<" Odd Parity:";

for(int i=1;i<=data\_bits+parity\_bits;i++)

{

cout<<code\_sent[i];

}

cout<<endl;

}

void get\_received()

{

for(int i=1;i<=data\_bits+parity\_bits;i++)

code\_received[i]=code\_sent[i];

}

void disturbance()

{

srand(time(0));

int i = rand()%(data\_bits+parity\_bits)+1;

if(code\_received[i]==1)

code\_received[i]=0;

else

code\_received[i]=1;

}

void print\_received()

{

cout<<"Code received with:";

if(parity=='E')

cout<<" Even Parity:";

else

cout<<" Odd Parity:";

for(int i=1;i<=data\_bits+parity\_bits;i++)

cout<<code\_received[i];

}

bool equal()

{

for(int i=1;i<=data\_bits+parity\_bits;i++)

if(code\_sent[i]!=code\_received[i])

return false;

return true;

}

void error\_checking()

{

int p=0;

int min,max=0,bit\_sum,j,k;

int code[10]={0};

int q=1;

for(int i=1;i<=data\_bits+parity\_bits;i=pow(2,p))

{

p++;

bit\_sum=0;

j=i;

k=i;

min=1;

max=i;

for(j;j<=data\_bits+parity\_bits;)

{

for(k=j;max>=min&&k<=data\_bits+parity\_bits;++min,++k)

{

if(code\_received[k]==1)

bit\_sum++;

}

j=k+i;

min=1;

}

if(parity=='E')

{

if(bit\_sum%2==0)

code[q]=0;

else

code[q]=1;

}

else

{

if(bit\_sum%2!=0)

code[q]=0;

else

code[q]=1;

}

q++;

}

int error=0;

int p1=0;

for(int l=1,p1=0;l<q;l++,p1++)

{

error+=code[l]\*pow(2,p1);

}

cout<<"Error is in bit no.:"<<error<<endl;

}

};

int main()

{

int choice;

Hamming\_code obj;

do

{

cout<<"1.Sender Side\n2.Receiver Side\n3.Exit\n";

cin>>choice;

switch(choice)

{

case 1:

cout<<"1.Without Disturbance\n2.With Disturbance\n";

cin>>choice;

obj.cal\_pbits();

obj.read\_message();

obj.encode\_msg();

obj.get\_received();

switch(choice)

{

case 1:

obj.sent\_print();

break;

case 2:

obj.disturbance();

obj.sent\_print();

break;

}

break;

case 2:

obj.print\_received();

if(obj.equal())

cout<<"\nNo Error in received code."<<endl;

else

obj.error\_checking();

break;

}

}while(choice!=3);

}

**Ouput:**

1.Sender Side

2.Receiver Side

3.Exit

1

1.Without Disturbance

2.With Disturbance

1

Enter type of Parity(E/O):O

Enter number of data bits:4

No of parity bits: 3

Total number of data bits:7

Enter Message:0 1 0 1

Message entered is:0101

Code Sent with Odd Parity:1001101

1.Sender Side

2.Receiver Side

3.Exit

2

Code received with: Odd Parity:1001101

No Error in received code.

1.Sender Side

2.Receiver Side

3.Exit

1

1.Without Disturbance

2.With Disturbance

2

Enter type of Parity(E/O):E

Enter number of data bits:7

No of parity bits: 4

Total number of data bits:11

Enter Message:1 1 0 1 0 0 1

Message entered is:1101001

Code Sent with Even Parity:01101011001

1.Sender Side

2.Receiver Side

3.Exit

2

Code received with: Even Parity:01101010001

Error is in bit no.:8

1.Sender Side

2.Receiver Side

3.Exit