***Experiment 04: Checksum***

***Sender Side Code:***

#include<bits/stdc++.h>

using namespace std;

class binary\_utilities

{

public:

///Add two binary numbers

vector<int> add(vector<int> b1,vector<int> b2,int b\_size)

{

int itr=b\_size-1;

vector<int> res(b\_size,0);

int c=0;

while(itr>=0)

{

///Get bits

int b1\_bit=b1[itr];

int b2\_bit=b2[itr];

///If both 1

if(b1\_bit==1 && b2\_bit==1)

{

///Carry wa 01

if(c==1)

{

c=1;

res[itr]=1;

}

///Carry was not 01

else

{

c=1;

res[itr]=0;

}

}

///Any one bit is 01 case

else if(b1\_bit==1 || b2\_bit==1)

{

///Carry wa 01

if(c==1)

{

c=1;

res[itr]=0;

}

///Carry was not 01

else

{

res[itr]=1;

}

}

///Both bits zero case

else

{

///Carry wa 01

if(c==1)

{

c=0;

res[itr]=1;

}

///Carry was not 01

else

{

res[itr]=0;

}

}

///Update itr

itr--;

}

///iF CARRY = 1 => rEADD IT TO ANS OBTAINED

if(c==1)

{

vector<int> temp(b\_size,0);

temp[b\_size-1]=1;

return add(res,temp,b\_size);

}

return res;

}

///Complement given binary number

vector<int> complementor(vector<int> a)

{

vector<int> comp;

for(int i=0; i<a.size(); i++)

{

comp.push\_back(a[i]==1?0:1);

}

return comp;

}

///Process transalte he gien daa stream into blocks

/// It transalted into blocks and added

vector<int> process(int n,int b\_size,vector<int> data)

{

vector<int> prev,curr,comp;

///Init prev and curr

int itr1=0;

int itr2=b\_size;

bool init=false;

int additions=0;

while(itr2<data.size())

{

///If not initialsed

if(!init)

{

while(itr2<(2\*b\_size))

{

prev.push\_back(data[itr1++]);

curr.push\_back(data[itr2++]);

init=true;

}

}

///prev will be result of prev addition

///Curent will be the next block

prev=add(prev,curr,b\_size);

int ele=0;

///Clean the curr

curr.erase(curr.begin(),curr.end());

///Fill in the new curr

while(itr2<data.size() && ele<b\_size)

{

curr.push\_back(data[itr2++]);

ele++;

}

additions++;

}

if(additions!=(data.size()/b\_size)-1) return add(prev,curr,b\_size);

return prev;

}

};

class checksumSender: public binary\_utilities

{

vector<int> data;

vector<int> codeword;

int b\_size;

int no\_of\_bits;

public:

checksumSender(int n,int b\_size,vector<int> data)

{

this->b\_size=b\_size;

this->no\_of\_bits=n;

this->data=data;

sender();

}

/// Sender

void sender()

{

///Get addition's complement

vector<int> checksum\_word=process(no\_of\_bits,b\_size,data);

///process will retun sum of blocks

///What wil be sent receiver is data and complement

checksum\_word=complementor(checksum\_word);

///Checksum generated is:

cout<<"Checksum is ";

for(int i=0;i<checksum\_word.size();i++){

cout<<checksum\_word[i]<<" ";

}

///Append Complemented block to data to make code word

for(int i=0;i<checksum\_word.size();i++){

data.push\_back(checksum\_word[i]);

}

cout<<endl;

cout<<"Data sent to receiver [Data with checksum] is: ";

for(int i=0;i<data.size();i++) cout<<data[i]<<" ";

cout<<endl;

}

};

int main()

{

int n,b\_size,bit;

vector<int> data;

cout<<"Enter the number of bits in data stream: ";

cin>>n;

cout<<"Enter the block size: ";

cin>>b\_size;

cout<<"Enter the data stream to be sent ";

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

{

cin>>bit;

data.push\_back(bit);

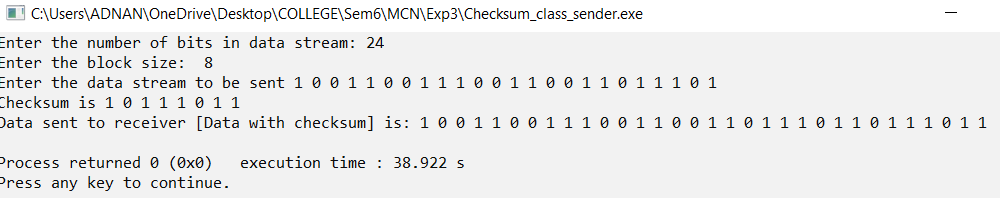
}

checksumSender c1=checksumSender(n,b\_size,data);

return 0;

}

***Output:***

******

***Receiver Side Code:***

#include<bits/stdc++.h>

using namespace std;

class binary\_utilities

{

public:

///Add two binary numbers

vector<int> add(vector<int> b1,vector<int> b2,int b\_size)

{

int itr=b\_size-1;

vector<int> res(b\_size,0);

int c=0;

while(itr>=0)

{

///Get bits

int b1\_bit=b1[itr];

int b2\_bit=b2[itr];

///If both 1

if(b1\_bit==1 && b2\_bit==1)

{

///Carry wa 01

if(c==1)

{

c=1;

res[itr]=1;

}

///Carry was not 01

else

{

c=1;

res[itr]=0;

}

}

///Any one bit is 01 case

else if(b1\_bit==1 || b2\_bit==1)

{

///Carry wa 01

if(c==1)

{

c=1;

res[itr]=0;

}

///Carry was not 01

else

{

res[itr]=1;

}

}

///Both bits zero case

else

{

///Carry wa 01

if(c==1)

{

c=0;

res[itr]=1;

}

///Carry was not 01

else

{

res[itr]=0;

}

}

///Update itr

itr--;

}

///iF CARRY = 1 => rEADD IT TO ANS OBTAINED

if(c==1)

{

vector<int> temp(b\_size,0);

temp[b\_size-1]=1;

return add(res,temp,b\_size);

}

return res;

}

///Complement given binary number

vector<int> complementor(vector<int> a)

{

vector<int> comp;

for(int i=0; i<a.size(); i++)

{

comp.push\_back(a[i]==1?0:1);

}

return comp;

}

///Process transalte he gien daa stream into blocks

/// It transalted into blocks and added

vector<int> process(int n,int b\_size,vector<int> data)

{

vector<int> prev,curr,comp;

///Init prev and curr

int itr1=0;

int itr2=b\_size;

bool init=false;

int additions=0;

while(itr2<data.size())

{

///If not initialsed

if(!init)

{

while(itr2<(2\*b\_size))

{

prev.push\_back(data[itr1++]);

curr.push\_back(data[itr2++]);

init=true;

}

}

///prev will be result of prev addition

///Curent will be the next block

prev=add(prev,curr,b\_size);

int ele=0;

///Clean the curr

curr.erase(curr.begin(),curr.end());

///Fill in the new curr

while(itr2<data.size() && ele<b\_size)

{

curr.push\_back(data[itr2++]);

ele++;

}

additions++;

}

if(additions!=(data.size()/b\_size)-1) return add(prev,curr,b\_size);

return prev;

}

};

class checksumReceiver: public binary\_utilities

{

vector<int> data;

vector<int> codeword;

int b\_size;

int no\_of\_bits;

bool e=false; ///error

public:

checksumReceiver(int n,int b\_size,vector<int> data,bool e)

{

this->b\_size=b\_size;

this->no\_of\_bits=n;

this->data=data;

this->e=e;

receiver();

}

///Messaeg displayer

void display(bool error)

{

if(error)

{

cout<<"\n- - - - - - - - - - - - - - - - -";

cout<<"\nE R R O R \_\_ D E T E C T E D\nA C K \_\_ T Y P E: negative\nM E S S A G E : Please retransmit the data";

error=true;

cout<<"\n- - - - - - - - - - - - - - - - -";

}

else

{

cout<<"\n- - - - - - - - - - - - - - - - -";

cout<<"\nNO \_\_ E R R O R \_\_ D E T E C T E D\nA C K \_\_ T Y P E: positive\nM E S S A G E : Data receivd successfully";

cout<<"\n- - - - - - - - - - - - - - - - -";

cout<<"\nActual data extracted from code word is: ";

int last=data.size()/b\_size;

for(int i=0; i<data.size()-b\_size; i++)

{

cout<<data[i]<<" ";

}

}

}

/// Sender

void receiver()

{

///Phase where error is introduce

if(e)

{

data[1]=data[1]==1?0:1;

}

cout<<endl;

cout<<"Data Received from sender is: ";

for(int i=0; i<data.size(); i++)

{

cout<<data[i]<<" ";

}

cout<<endl;

///Get addition's complement from current data,data has checksum so it too will be added along

vector<int> comp\_check=process(no\_of\_bits,b\_size,data);

cout<<"the sum obtained after adding blocks is: ";

for(int i=0; i<comp\_check.size(); i++)

{

cout<<comp\_check[i]<<" ";

}

cout<<endl;

comp\_check=complementor(comp\_check);

cout<<"Complement of sum is: ";

for(int i=0; i<comp\_check.size(); i++)

{

cout<<comp\_check[i]<<" ";

}

cout<<endl;

bool error=false;

for(int i=0; i<comp\_check.size(); i++)

{

if(comp\_check[i]==1)

{

error=true;

display(error);

break;

}

}

if(!error)

{

display(error);

}

}

};

int main()

{

int n,b\_size,bit;

vector<int> data;

cout<<"Enter the number of bits in code word received: ";

cin>>n;

cout<<"Enter the block size: ";

cin>>b\_size;

cout<<"Enter the data stream received: ";

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

{

cin>>bit;

data.push\_back(bit);

}

checksumReceiver c1=checksumReceiver(n,b\_size,data,false);

cout<<endl;

checksumReceiver c2=checksumReceiver(n,b\_size,data,true);

return 0;

}

***Output:***

1. ***Case where No Error exists.***
2. ***Case where Error is found.***

******