Name : Kumbhani Sanket

Div : A

Roll_no : 3111

DCN[P]_Assignment-3

Program: 1

```
//Kumbhani Sanket - 3111
import java.io.*;
import java.net.*;
import java.util.Scanner;

class server_checksum
{
	public static void main(String[] args) throws Exception
	{
		try
		{
			ServerSocket serversocket = new ServerSocket(6666);
			System.out.println("server is listening on
localhost:6666");

			Socket socket = serversocket.accept();
```

```
                DataInputStream instream = new
DataInputStream(socket.getInputStream());
                DataOutputStream ostream = new
DataOutputStream(socket.getOutputStream());

                String rmsg = instream.readUTF();
                System.out.println("\nReceive msg ="+rmsg);
                int len = 4;
                // System.out.println(len);
                String result = "";

                while(result.length()<len)
                {
                        result="0"+result;
                }

                for(int i=0;i<rmsg.length();i+=len)
                {
                        String temp = rmsg.substring(i,i+len);
                        result = binadd(result,temp);
                        // System.out.println("result = "+result);
                }
                result = ones(result);
                System.out.println("final result = "+result);

                int flag=0;
                for(int i=0;i<result.length();i++)
                {
                        if(result.charAt(i)!='0')
                                System.out.println("Error");
                        else
                                flag=1;
                }
```

```
                if(flag!=0)
                        System.out.println("Run Successfully.");

                ostream.close();
                instream.close();
                socket.close();
                serversocket.close();
        }
        catch(Exception e)
        {
                System.out.println(e);
        }
    }


public static String binadd(String a,String b)
    {
        String result = "";
        String carry = "0";
        for(int i=a.length()-1;i>=0;i--)
        {

                if(a.charAt(i)==b.charAt(i))
                {
                        if(a.charAt(i)=='1')
                        {
                                if(carry == "0")
                                {
                                        result = "0"+result;
                                        carry="1";
                                }
                                else
                                {
```

```
                        result = "1"+result;
                        carry="1";
                }
        }
        else
        {
                if(carry=="1")
                {
                        result = "1"+result;
                        carry="0";
                }
                else
                {
                        result = "0"+result;
                        carry="0";
                }
        }

}
else
{
        if(carry=="1")
        {
                        result = "0" + result;
                        carry="1";
        }
        else
        {
                result = "1"+ result;
                carry="0";
        }
}
}
```

```
            if(carry == "1")
            {
                while(carry.length()<a.length())
                    carry="0"+carry;
                // System.out.print("carry encounter = ");
                result = binadd(carry,result);
            }
            return result;
        }

        public static String ones(String msg)
        {
            String ans = "";
            for(int i=0;i<msg.length();i++)
            {
                if(msg.charAt(i)=='0')
                {
                    ans = ans+"1";
                }
                else
                {
                    ans = ans+"0";
                }
            }
            return ans;
        }
}
```

```
//Kumbhani Sanket - 3111
import java.io.*;
import java.net.*;
```

```
import java.util.Scanner;

class client_checksum
{
	public static void main(String[] args) throws Exception
	{
		try
		{
			Socket socket = new Socket("localhost",6666);

			DataOutputStream ostream = new
DataOutputStream(socket.getOutputStream());
			DataInputStream instream = new
DataInputStream(socket.getInputStream());

			Scanner sc = new Scanner(System.in);
			System.out.print("Enter msg:");
			String msg = sc.nextLine();
			System.out.print("Enter length:");
			// int len = sc.nextInt();
			// ostream.writeInt(len);
			int len = 4;

			String result = new String();
			while(result.length()<len)
			{
				result="0"+result;
			}

			while(msg.length()%len!=0)
			{
				msg="0"+msg;
			}
```

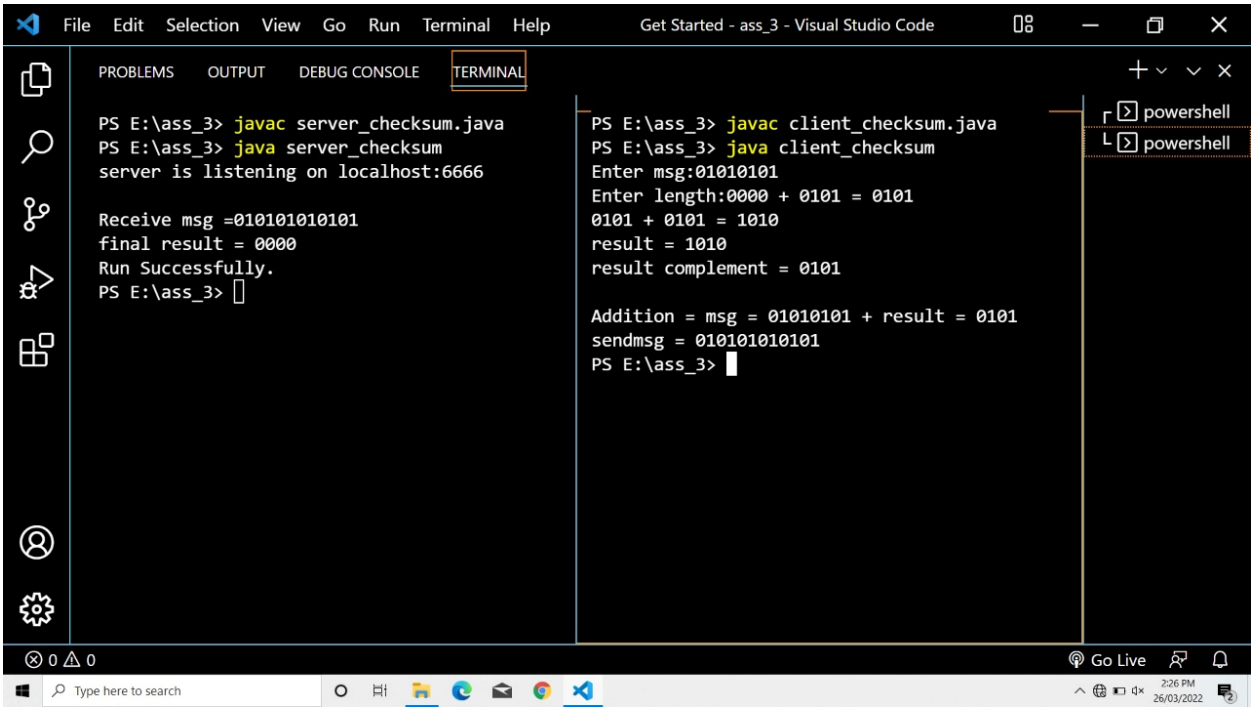
```
                    for(int i=0;i<msg.length();i+=len)
                    {
                        String temp = msg.substring(i,i+len);
                        System.out.print(result+" + "+temp+" = ");
                        result=binadd(result,temp);
                        System.out.println(result);
                        // System.out.println("result = "+result);
                    }
                    System.out.println("result = "+result);
                    result = ones(result);
                    System.out.println("result complement = "+result);
                    System.out.println("\nAddition = msg = "+msg+" +
result = "+result);
                    String sendmsg = msg + result;

                    ostream.writeUTF(sendmsg);
                    ostream.flush();
                    System.out.println("sendmsg = "+sendmsg);

                    instream.close();
                    ostream.close();
                    socket.close();
                }
                catch(Exception e)
                {
                    System.out.println(e);
                }
        }


        public static String binadd(String a,String b)
        {
            String result = "";
```

```
String carry = "0";
for(int i=a.length()-1;i>=0;i--)
{

        if(a.charAt(i)==b.charAt(i))
        {
                if(a.charAt(i)=='1')
                {
                        if(carry == "0")
                        {
                                result = "0"+result;
                                carry="1";
                        }
                        else
                        {
                                result = "1"+result;
                                carry="1";
                        }
                }
                else
                {
                        if(carry=="1")
                        {
                                result = "1"+result;
                                carry="0";
                        }
                        else
                        {
                                result = "0"+result;
                                carry="0";
                        }
                }
```

```
                }
                else
                {
                        if(carry=="1")
                        {
                                result = "0" + result;
                                carry="1";
                        }
                        else
                        {
                                result = "1"+ result;
                                carry="0";
                        }
                }
        }
        if(carry == "1")
        {
                while(carry.length()<a.length())
                        carry="0"+carry;
                // System.out.print("carry encounter = ");
                result = binadd(carry,result);
        }
        return result;
}

public static String ones(String msg)
{
        String ans = "";
        for(int i=0;i<msg.length();i++)
        {
                if(msg.charAt(i)=='0')
                {
                        ans = ans+"1";
```

```
                }
                else
                {
                        ans = ans+"0";
                }
            }
            return ans;
        }
}
```

Output:



Program: 2

```
//Kumbhani Sanket-3111
import java.net.*;
import java.io.*;
import java.util.Scanner;

class receiver_crc
{
   public static void main(String args[]) throws Exception
   {
      try{
         DatagramSocket ds1 = new DatagramSocket(6363);
         System.out.println("server listening on localhost:6363");
         byte[] buf = new byte[500];
         DatagramPacket dp1 = new DatagramPacket(buf,500);
         ds1.receive(dp1);
         ds1.close();

         String data = new String(dp1.getData(),0,dp1.getLength());
         System.out.println("Received msg = "+ data);
         String key = "11";
         String rem = div(data,key);
         System.out.println("rem = "+rem);
         int cnt=0;
         for(int i=0;i<rem.length();i++)
         {
            if(rem.charAt(i)=='0')
               cnt++;
         }
         if(cnt!=0)
            System.out.println("No error -> Successful.");

         // DatagramSocket ds2 = new DatagramSocket();
         // Scanner s = new Scanner(System.in);
```

```
        // System.out.println("server:");
        // String msg1 = s.nextLine();
        // InetAddress ip = InetAddress.getByName("localhost");
        // DatagramPacket dp2 = new
DatagramPacket(msg1.getBytes(),msg1.length(),ip,6565);
        // ds2.send(dp2);
      }catch(Exception e)
      {System.out.println(e);}
   }

   public static String xor(String a, String b)
   {
      String result = "";
      for(int i=1;i<a.length();i++)
      {
         if(a.charAt(i) == b.charAt(i))
            result = result + "0";
         else
            result = result + "1";
      }
      return result;
   }

   public static String div(String data, String key)
   {
      int len = key.length();
      for(int i=0;i<len-1;i++)
      {
         data = data + "0";
      }

      String zero = "";
      for(int i=0;i<len;i++)
```

```
      {
         zero = zero + "0";
      }

      String temp = data.substring(0,len);
      while(len<data.length())
      {
         if(temp.charAt(0)=='1')
            temp = xor(temp,key) + data.charAt(len);
         else
            temp = xor(temp,zero) + data.charAt(len);
         len++;
      }

      if(temp.charAt(0)=='1')
         temp = xor(temp,key);
      else
         temp = xor(temp,zero);

      return temp;
   }
}
```

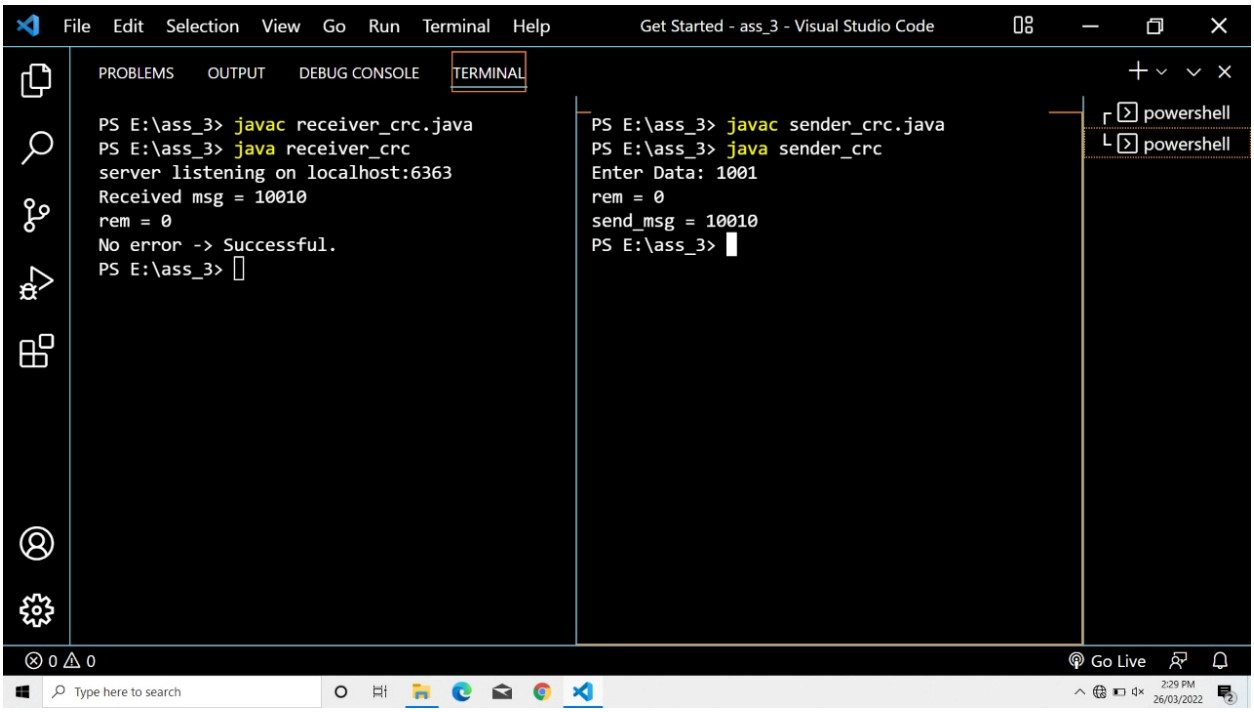
```
//Kumbhani Sanket-3111
import java.net.*;
import java.io.*;
import java.util.Scanner;

class sender_crc
{
   public static void main(String args[]) throws Exception
```

```
  {
    try{
      DatagramSocket ds1 = new DatagramSocket();
      Scanner sc = new Scanner(System.in);
      System.out.print("Enter Data: ");
      String data = sc.nextLine();
      String key = "11";
      String rem = div(data,key);
      System.out.println("rem = "+rem);
      String sendmsg = data + rem;
      System.out.println("send_msg = "+sendmsg);

      InetAddress ip = InetAddress.getByName("localhost");

      DatagramPacket dp1 = new
DatagramPacket(sendmsg.getBytes(),sendmsg.length(),ip,6363);
      ds1.send(dp1);

    }catch(Exception e)
    {System.out.println(e);}
  }

  public static String xor(String a, String b)
  {
    String result = "";
    for(int i=1;i<a.length();i++)
    {
      if(a.charAt(i) == b.charAt(i))
        result = result + "0";
      else
        result = result + "1";
    }
    return result;
```

```
}

public static String div(String data, String key)
{
   int len = key.length();
   for(int i=0;i<len-1;i++)
   {
      data = data + "0";
   }

   String zero = "";
   for(int i=0;i<len;i++)
   {
      zero = zero + "0";
   }

   String temp = data.substring(0,len);
   while(len<data.length())
   {
      if(temp.charAt(0)=='1')
         temp = xor(temp,key) + data.charAt(len);
      else
         temp = xor(temp,zero) + data.charAt(len);
      len++;
   }

   if(temp.charAt(0)=='1')
      temp = xor(temp,key);
   else
      temp = xor(temp,zero);

   return temp;
}
```

}

Output:



Program: 3

```
//Kumbhani Sanket-3111
import java.net.*;
import java.io.*;
import java.util.Scanner;
import java.lang.Math;
import java.util.Arrays;

class receiver_hamming
{
   public static void main(String args[]) throws Exception
   {
      try{
```

```
DatagramSocket ds1 = new DatagramSocket(6363);
System.out.println("server listening on localhost:6363");
byte[] buf = new byte[500];
DatagramPacket dp1 = new DatagramPacket(buf,500);
ds1.receive(dp1);
ds1.close();

String data = new String(dp1.getData(),0,dp1.getLength());
System.out.println("Received msg = "+ data);
// int a = Integer.parseInt(data);
// System.out.println(a);
int totalbits = data.length();
int rbits = 3;
int databits = totalbits - rbits;
// System.out.println(totalbits);
int rhcode[] = new int[totalbits+1];

//storing data into array
for(int i=0;i<totalbits;i++)
{
   rhcode[i] = data.charAt(i) - '0';
}
System.out.println("***** Received Code : *****");
for(int i=0;i<totalbits;i++)
{
   System.out.print("rhcode["+i+"] : "+rhcode[i]+"\t");
}
System.out.println();

//create array to store errorbits
int errorindex[] = new int[rbits];

for(int i=1,x=0,e=0; i<=totalbits; i++)
```

```
        {
           if(Math.pow(2,x)==i)
           {
              int counter = 0;

              for(int sindex=i; sindex<=totalbits; sindex=sindex+i+i)
              {
                    for(int index=sindex,cinc=1; index<=totalbits &&
cinc<=i; index++,cinc++)
                    {
                          //System.out.println("p["+index+"] :
"+hcode[index]);
                          if(rhcode[index] == 1)
                          {
                             counter++;
                          }
                    }
              }
              if(counter % 2 != 0)
              {
                 errorindex[e]=i;
                 e++;
              }
              x++;
           }
        }

        //check if any error is there or not
        int sum=0;
        System.out.print("\n errorIndex array values:");
        for(int i=0;i<rbits;i++)
        {
           System.out.print(" "+ errorindex[i]);
```

```
   sum = sum + errorindex[i];
}

//if sum < 0 then no error else error
if(sum>0)
{
   System.out.println("\n Error at index: "+ sum);
   //correct error, flip the bit at error index (0 to 1) & (1 to 0)
   if(rhcode[sum]==0)
      rhcode[sum]=1;
   else
      rhcode[sum]=0;
}
else
{
   System.out.println("\nNo error in hamming code.");
}

System.out.print("\n final received hamming code: ");
for(int i=totalbits; i>=1; i--)
{
   System.out.print(rhcode[i]);
}
System.out.println();

//extarct data from hamming code
int rdata[] = new int[databits+1];
for(int i=1,x=1,d=0; i<=totalbits; i++)
{
   if(Math.pow(2,x)==i)
   {
      x++;
   }
```

```
            else
            {
               rdata[d] = rhcode[i];
               d++;
            }
         }

         System.out.print("\n received Data: ");
         for(int i = databits; i>=1; i--)
         {
            System.out.print(rdata[i]);
         }
         System.out.println();
      }catch(Exception e)
      {System.out.println(e);}
   }

}
```

```
//Kumbhani Sanket-3111
import java.net.*;
import java.io.*;
import java.util.Scanner;
import java.lang.Math;

class sender_hamming
{
   public static void main(String args[]) throws Exception
   {
      try{
         DatagramSocket ds1 = new DatagramSocket();
```

```
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Length of data:");
        int databits = sc.nextInt();

        int rbits = 0;
        System.out.println("\nCalculating rbits:");
        while(Math.pow(2,rbits) < databits+rbits+1)
        {
           System.out.println("2^" + rbits + " < " +databits+" + "+rbits+" +
1");
           rbits++;
        }

        int totalbits = databits + rbits;
        System.out.println("\nData bits: "+databits);
        System.out.println("Redundent Bits: "+rbits);
        System.out.println("Total Bits: "+totalbits);

        //create array and store the data
        int data[] = new int[databits+1];

        //data stored in reverse order
        System.out.println("\nEnter data into array:");
        for(int i=databits; i>=1; i--)
        {
           data[i] = sc.nextInt();
        }
        System.out.println();

        //printing array
        System.out.println("\n *****data array:*****");
        for(int i=1;i<=databits; i++)
        {
```

```
   System.out.print("\tdata["+i+"] = "+data[i]);
}
System.out.println();

//create array & store hamming code
int hcode[] = new int[totalbits+1];

//set values in hamming code
for(int i=1, x=0, k=1; i<=totalbits; i++)
{
      if(Math.pow(2, x) == i)
      {
         hcode[i] = 0;
         x++;
      }
      else
      {
         hcode[i]=data[k];
         k++;
      }
}
//printing hcode array
System.out.println("\n *****hcode array:*****");
for(int i=1;i<=totalbits;i++)
{
   System.out.print("\thcode["+i+"] = "+hcode[i]);
}
System.out.println();

//calculating value of rbits
for(int i=1,x=0; i<=totalbits;i++)
{
   if(Math.pow(2,x)==i)
```

```
            {
               int counter = 0;
               System.out.println("\nrbits index : "+i);

               for(int sindex=i; sindex<=totalbits; sindex=sindex+i+i)
               {
                     for(int index=sindex,cinc=1; index<=totalbits &&
cinc<=i; index++,cinc++)
                     {
                           System.out.print("\tp["+index+"] : "+hcode[index]);
                           if(hcode[index] == 1)
                           {
                              counter++;
                           }
                     }
               }
               if(counter % 2 != 0)
               {
                  hcode[i]=1;
               }
               else
               {
                  hcode[i]=0;
               }
               x++;
            }
         }

         //print final hamming code
         System.out.print("\n hamming code:\t");
         String store = "";
         for(int i=totalbits; i>=1; i--)
         {
```

```
            System.out.print(hcode[i]);
            store += Integer.toString(hcode[i]);
        }
        System.out.println();

        InetAddress ip = InetAddress.getByName("localhost");

        DatagramPacket dp1 = new
DatagramPacket(store.getBytes(),store.length(),ip,6363);
        ds1.send(dp1);

    }catch(Exception e)
    {System.out.println(e);}
  }

}
```

Output: