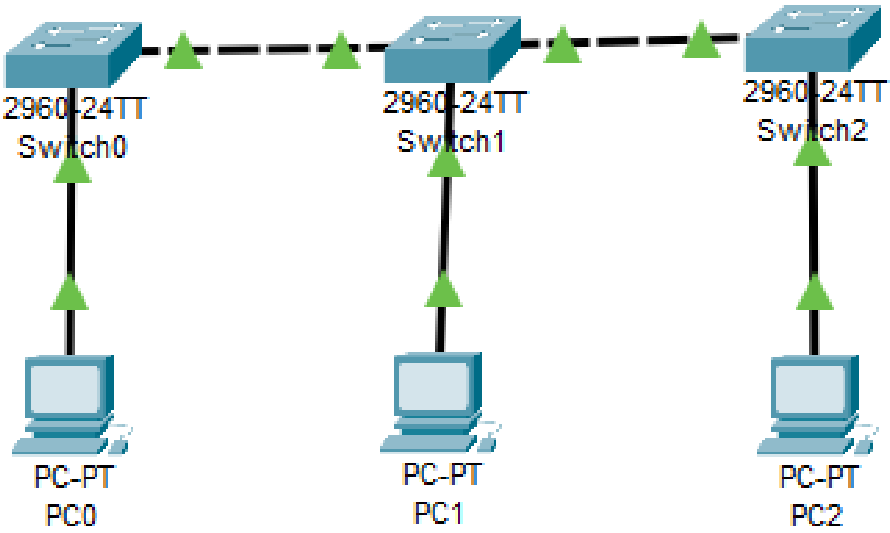
1. a) Write a program to implement the Data link layer framing method bit stuffing.

```
import java.util.*;
public class Bit
{
	public static void main(String[] args)
	{
		int i,count=0;
		Scanner str=new Scanner(System.in);
		System.out.println("Enter bits : ");
		String s1=str.nextLine();

		for(i=0;i<s1.length();i++)
		{
			if(s1.charAt(i)=='1')
				count++;
			if(s1.charAt(i)=='0')
				count=0;
			System.out.print(s1.charAt(i));
			if(count==5)
			{
				System.out.print("0");
				count=0;
				str.close();
			}
		}
	}
}
```

OUTPUT

```
Enter bits : 101111111
1011111011
```

b) Configure bus topology using cisco packet tracer.



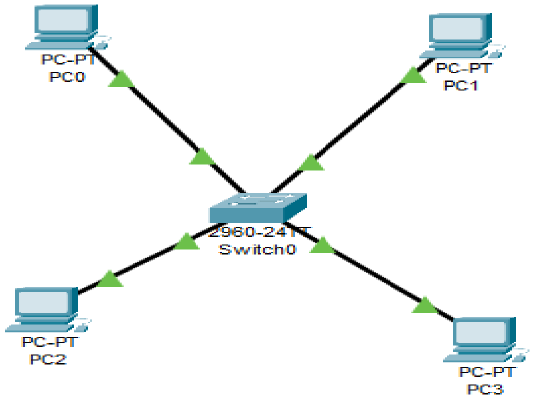
2. a) Write a program to implement the Data link layer framing method character stuffing.

```
import java.util.*;
public class Char
{
      public static void main(String[] args)
      {
            Scanner sc =new Scanner(System.in);
            System.out.println("Enter number of Characters : ");
            int n=sc.nextInt();
            String in[]=new String[n];
            for(int i=0;i<n;i++)
            {
                  in[i]=sc.next();
            }
            for(int i=0;i<n;i++)
            {
                  if(in[i].equals("DLE"))
                  {
                        in[i]="DLE DLE";
                  }
            }
            System.out.print("Transmitted Message is  "+"DLE STX ");
            for(int i=0;i<n;i++)
            {
                  System.out.print(in[i]+" ");
            }
            System.out.print("DLE ETX");
            sc.close();
            int i = 0;
            System.out.println("\nOriginal Message : "+in[i]);
      }
}
```

OUTPUT

```
Enter number of Characters : 1
DOODLE
Transmitted Message is  DLE STX DOODLE DLE ETX
Original Message : DOODLE
```

b) Configure star topology using cisco packet tracer.



3. Write a program to simulate Stop and wait protocol.

mysender.java

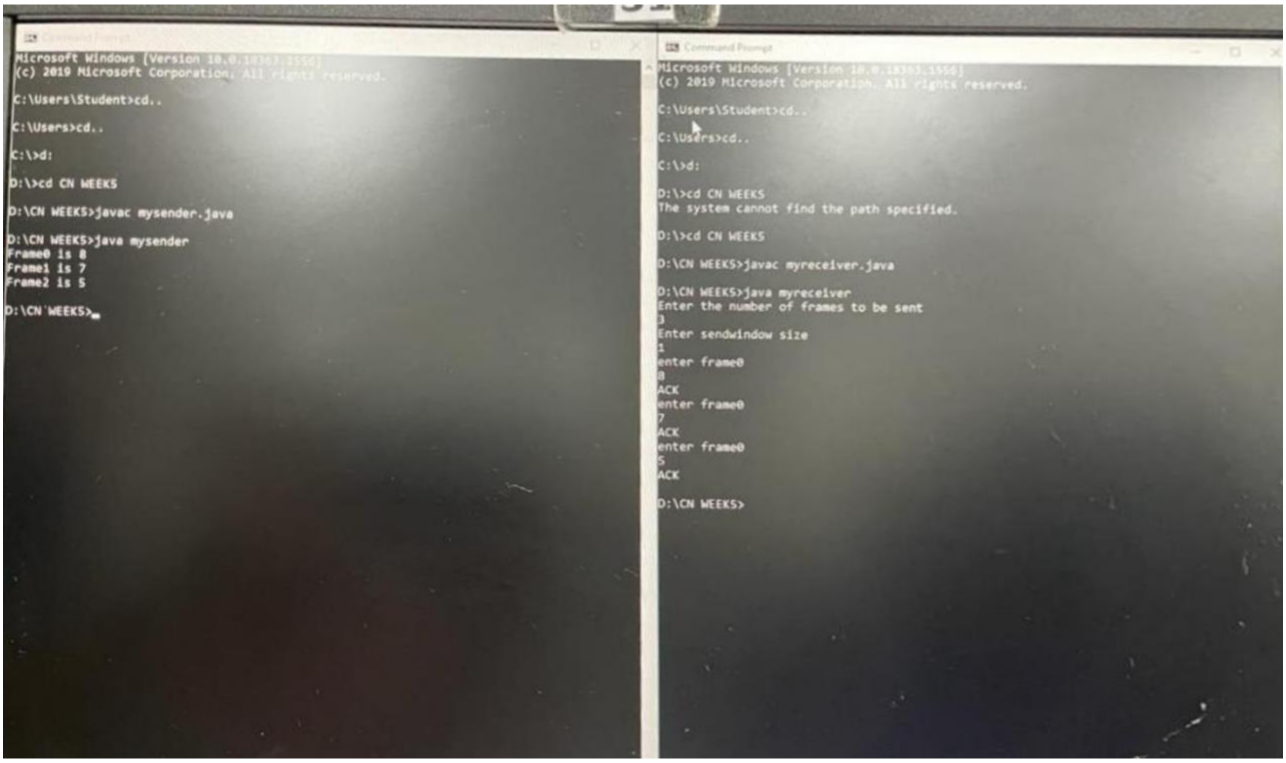
```
import java.io.*;
import java.net.*;
public class mysender {
      public static void main(String args[])
      {
            try {
                  ServerSocket ss=new ServerSocket(113);
                  Socket s=ss.accept();
DataInputStream dis=new DataInputStream(s.getInputStream());
DataOutputStream dos=new DataOutputStream(s.getOutputStream());
                  int p=Integer.parseInt(dis.readUTF());
                  //intreceivewindow=1;
                  for(int i=0;i<p;i++)
                  {
                  String Str[]=new String[p];
                  Str[i]=(String)dis.readUTF();
                  System.out.println("Frame"+i+" is "+Str[i]);
                  //System.out.println("Ack sent");
                  dos.writeUTF("ACK");
                  }
                  ss.close();
            }catch(Exception e) {System.out.print(e);}
      }
}
```

Myreceiver.java

```
import java.io.*;
import java.net.*;
import java.util.*;
public class myreceiver {
      public static void main(String args[])
      {
            try {
                  Socket s=new Socket("localhost",113);
      DataOutputStream dout=new DataOutputStream(s.getOutputStream());
      DataInputStream di=new DataInputStream(s.getInputStream());
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter the number of frames to be sent");
                  int n=sc.nextInt();
                  String k= Integer.toString(n);
                  dout.writeUTF(k);
                  String ack[]=new String[n];
                  int sendwindow=0;
                  System.out.println("Enter sendwindow size");
                  sendwindow=sc.nextInt();
                  for(int i=0,j=0;i<n/sendwindow;i++) {
                        while(j<sendwindow) {
                  System.out.println("enter frame"+j);
                  String fr=sc.next();
                  dout.writeUTF(fr);
                  ack[i]=(String)di.readUTF();
                  System.out.println(ack[i]);
                  j++;
```

```
                        }
                        j=0;
                  }
                  dout.flush();
                  dout.close();
                  s.close();
                  sc.close();
            }catch(Exception e) {System.out.println(e);}
      }

}
```

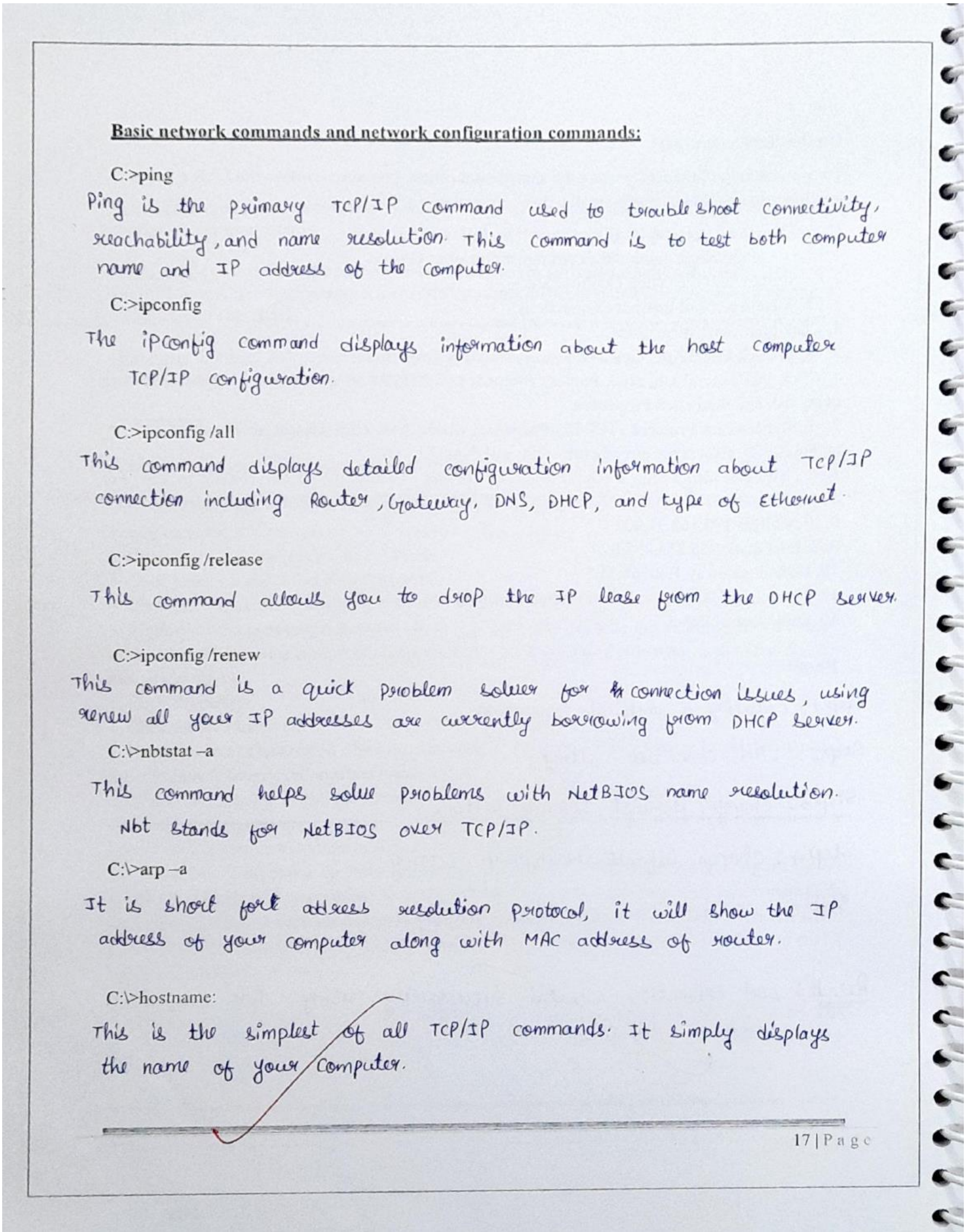
OUTPUT



```
Microsoft Windows [Version 10.0.18363.1556]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Student>cd..

C:\Users>cd..

C:\>d:

D:\>cd CN WEEKS

D:\CN WEEKS>javac mysender.java

D:\CN WEEKS>java mysender
Frame0 is 8
Frame1 is 7
Frame2 is 5

D:\CN WEEKS>_
```

```
Microsoft Windows [Version 10.0.18363.1556]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Student>cd..

C:\Users>cd..

C:\>d:

D:\>cd CN WEEKS
The system cannot find the path specified.

D:\>cd CN WEEKS

D:\CN WEEKS>javac myreceiver.java

D:\CN WEEKS>java myreceiver
Enter the number of frames to be sent
3
Enter sendwindow size
1
enter frame0
8
ACK
enter frame0
7
ACK
enter frame0
5
ACK

D:\CN WEEKS>
```

4. a) Explain about different types of basic network commands and implement them.



Basic network commands and network configuration commands:

C:>ping

Ping is the primary TCP/IP command used to troubleshoot connectivity, reachability, and name resolution. This command is to test both computer name and IP address of the computer.

C:>ipconfig

The ipconfig command displays information about the host computer TCP/IP configuration.

C:>ipconfig /all

This command displays detailed configuration information about TCP/IP connection including Router, Gateway, DNS, DHCP, and type of Ethernet.

C:>ipconfig /release

This command allows you to drop the IP lease from the DHCP server.

C:>ipconfig /renew

This command is a quick problem solver for connection issues, using renew all your IP addresses are currently borrowing from DHCP server.

C:\>nbtstat –a

This command helps solve problems with NetBIOS name resolution. Nbt stands for NetBIOS over TCP/IP.

C:\>arp –a

It is short fort address resolution protocol, it will show the IP address of your computer along with MAC address of router.

C:\>hostname:

This is the simplest of all TCP/IP commands. It simply displays the name of your computer.

C:\>ipconfig /flushdns:

This command is only needed if you are having trouble with networks DNS configuration. The best time to use is after network configuration.

C:\netdiag:

Netdiag is a network testing utility that performs a variety of network diagonstic tests, allowing to pinpoint problems in network.

C:\>netstat:

Netstat displays a variety of statistics about a computers active TCP/IP connections, it is used when having trouble with applications such as HTTP and FTP.

C:\>nslookup:

nslookup is used for diagnosing DNS problems. If you can access a resource by specifying an IP address.

C:\>pathping:

Pathping is unique to window's and is basically a combination of Ping and Tracert commands.

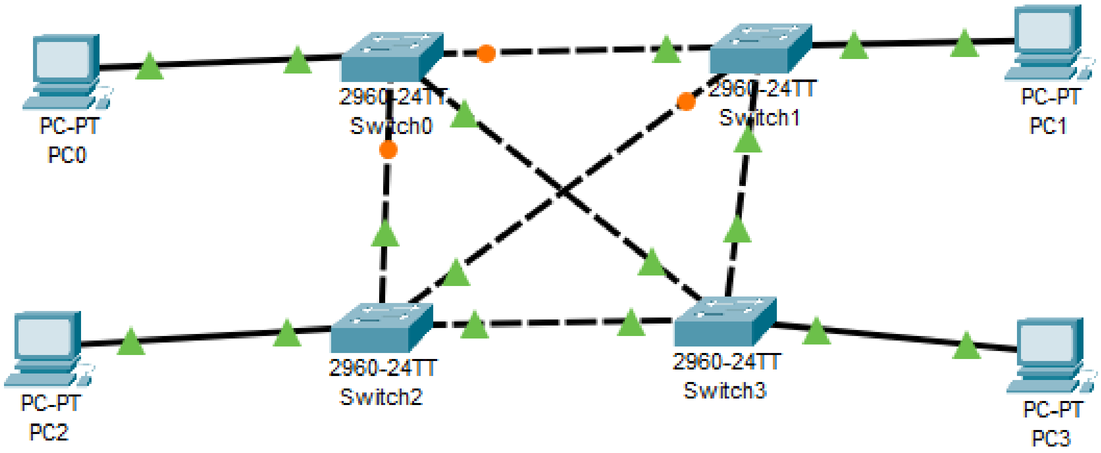
C:\>route:

The route command displays the computers routing table. Whenever having trouble accessing other computers on network, we can use the route command to make sure the Entries in routing table are correct.

C:\>tracert:

The tracert command displays a list of all routers that a packet has to go through to get from the computer where tracert is run to any other computer on the internet.

b) Configure mesh topology using cisco packet tracer.



5. Write a program to implement on a data set of characters using the three Cyclic Redundancy Check.

```
import java.util.*;
class crc1{public static void main(String args[]) {
Scanner scan = new Scanner(System.in);
int n;
System.out.println("Enter the size of the data:");
n = scan.nextInt();
int data[] = new int[n];
System.out.println("Enter the data, bit by bit:");
for(int i=0 ; i < n ; i++) {
System.out.println("Enter bit number " + (n-i) + ":");
data[i] = scan.nextInt();
}
System.out.println("Enter the size of the divisor:");
n = scan.nextInt();
int divisor[] = new int[n];
System.out.println("Enter the divisor, bit by bit:");
for(int i=0 ; i < n ; i++) {
System.out.println("Enter bit number " + (n-i) + ":");
divisor[i] = scan.nextInt();
}
int remainder[] = divide(data, divisor);
for(int i=0 ; i < remainder.length-1 ; i++) {
System.out.print(remainder[i]);
}
System.out.println("\nThe CRC code generated is:");
for(int i=0 ; i < data.length ; i++) {
System.out.print(data[i]);
}
for(int i=0 ; i < remainder.length-1 ; i++) {
System.out.print(remainder[i]);
}
System.out.println();
int sent_data[] = new int[data.length + remainder.length - 1];
System.out.println("Enter the data to be sent:");
for(int i=0 ; i < sent_data.length ; i++) {
System.out.println("Enter bit number " + (sent_data.length-i)
+ ":");
sent_data[i] = scan.nextInt();
}
receive(sent_data, divisor);
}
static int[] divide(int old_data[], int divisor[]) {
int remainder[] , i;
int data[] = new int[old_data.length + divisor.length];
System.arraycopy(old_data, 0, data, 0, old_data.length);
remainder = new int[divisor.length];
System.arraycopy(data, 0, remainder, 0, divisor.length);
for(i=0 ; i < old_data.length ; i++) {
System.out.println((i+1) + ".) First data bit is : "
+ remainder[0]);
System.out.print("Remainder : ");
if(remainder[0] == 1) {
for(int j=1 ; j < divisor.length ; j++) {
remainder[j-1] = exor(remainder[j], divisor[j]);
System.out.print(remainder[j-1]);
}
}
else {
for(int j=1 ; j < divisor.length ; j++) {
remainder[j-1] = exor(remainder[j], 0);
System.out.print(remainder[j-1]);
}
}
remainder[divisor.length-1] = data[i+divisor.length];
System.out.println(remainder[divisor.length-1]);
}
return remainder;
}
static int exor(int a, int b) {
if(a == b) {
return 0;
}
return 1;
}
static void receive(int data[], int divisor[]) {
int remainder[] = divide(data, divisor);
for(int i=0 ; i < remainder.length ; i++) {
if(remainder[i] != 0) {
System.out.println("There is an error in received data...");
return;
}
}
System.out.println("Data was received without any error.");
}
}
```

OUTPUT

```
Enter the size of the data:4
Enter the data, bit by bit

Enter bit number 4:1
Enter bit number 3:0
Enter bit number 2:1
Enter bit number 1:1

Enter the size of the divisor:4
Enter the divisor, bit by bit
Enter bit number 4:1
Enter bit number 3:0
Enter bit number 2:0
Enter bit number 1:1

1.) First data bit is : 1
Remainder : 0100
2.) First data bit is : 0
Remainder : 1000
3.) First data bit is : 1
Remainder : 0010
4.) First data bit is : 0
Remainder : 0100
010

The CRC code generated is:1011010
```

```
Enter the data to be sent
Enter bit number 7:1
Enter bit number 6:0
Enter bit number 5:1
Enter bit number 4:1
Enter bit number 3:0
Enter bit number 2:1
Enter bit number 1:0

1.) First data bit is : 1
Remainder : 0100
2.) First data bit is : 0
Remainder : 1001
3.) First data bit is : 1
Remainder : 0000
4.) First data bit is : 0
Remainder : 0000
5.) First data bit is : 0
Remainder : 0000
6.) First data bit is : 0
Remainder : 0000
7.) First data bit is : 0

Remainder : 0000
Data was received without any error.
```