2 D Parity Check Sender :

```
import java.io.*;
import java.net.*;
public class twoDParityS{
	public static void main(String args[]){
		String dataword = "101011110111010";

		// Split Dataword into equal no of bit.(Here 5 bit in each subdataword)

		String[] s = new String[3];

		s[0] = dataword.substring(0, 5);
		s[1] = dataword.substring(5, 10);
		s[2] = dataword.substring(10, 15);

		System.out.println("S0 : " + s[0]);
		System.out.println("S1 : " + s[1]);
		System.out.println("S2 : " + s[2]);

		System.out.println("Codewords are : ");

		//Calculate row wise parity.

		char p;
		for(int i=0; i<3; i++){
			p = pbg(s[i], "even");
			s[i] = s[i] + p;
			System.out.println(s[i]);
		}

		//Calculate column wise parity.
		String snext = "";
		String temp="";

		for(int i=0; i<6; i++){
			temp = s[0].charAt(i) + "" + s[1].charAt(i) + "" + s[2].charAt(i);
			p = pbg(temp, "even");
			snext = snext + p;
		}
		System.out.println(snext);

		String Codeword = s[0] + s[1] + s[2] + snext;

		/*if(codeword.charAt(5) == '1'){
			codeword.replace('0', 5);
		else
			codeword.replace('1', 5);*/
```

```
		System.out.println(""+s[0] + s[1] + s[2] + snext);
	}

	public static char pbg(String x, String p){
		int cnt = 0;
		char pb;
		for(int i = 0; i < x.length(); i++){
			if(x.charAt(i) == '1'){
				cnt++;
			}
		}
		if(p.equals("even")){
			if(cnt % 2 == 0){
				pb = '0';
			}
			else{
				pb = '1';
			}
		}
		else{
			if(cnt % 2 == 0){
				pb = '1';
			}
			else{
				pb = '0';
			}
		}
		return pb;
	}
}
```

2 D Parity Check Receiver :

```
import java.io.*;
import java.net.*;
public class twoDParityR{
	public static void main(String args[]){
		String dataword = "101011011010110101100100";

		// Split Dataword into equal no of bit.(Here 5 bit in each subdataword)

		String[] s = new String[4];

		s[0] = dataword.substring(0, 6);
		s[1] = dataword.substring(6, 12);
		s[2] = dataword.substring(12, 18);
		s[3] = dataword.substring(18, 24);
```

```
		System.out.println("S0 : " + s[0]);
		System.out.println("S1 : " + s[1]);
		System.out.println("S2 : " + s[2]);
		System.out.println("S3 : " + s[3]);

		System.out.println("Codewords are : ");

		//Calculate row wise parity.

		char p='0';
		for(int i=0; i<4; i++){
			p = pbg(s[i], "even");
			if(p != '0'){
				System.out.println("Data is Corrupted for row no : " + (i+1));
				break;
			}
			else{
				System.out.println("Data is error Free for row no : " + (i+1));
			}
		}

		//Calculate column wise parity.
		String snext = "";
		String temp="";

		for(int i=0; i<6; i++){
			temp = s[0].charAt(i) + "" + s[1].charAt(i) + "" + s[2].charAt(i)+ "" + s[3].charAt(i);
			p = pbg(temp, "even");
			if(p != '0'){
				System.out.println("Data is Corrupted for column no : " + (i+1));
				break;
			}
			else{
				System.out.println("Data is error Free for column no : " + (i+1));
			}
		}

		/*if(p == '0'){
			System.out.println("whole Data is Error Free.");
		}*/
	}

	public static char pbg(String x, String p){
		int cnt = 0;
		char pb;
		for(int i = 0; i < x.length(); i++){
			if(x.charAt(i) == '1'){
```

```
                        cnt++;
                }
        }
        if(p.equals("even")){
                if(cnt % 2 == 0){
                        pb = '0';
                }
                else{
                        pb = '1';
                }
        }
        else{
                if(cnt % 2 == 0){
                        pb = '1';
                }
                else{
                        pb = '0';
                }
        }
        return pb;
    }
}
```

Checksum Sender :

```
import java.io.*;
import java.net.*;
public class CheckSumS{
	public static void main(String args[]) throws Exception{
		/*Socket skt = new Socket("localhost", 6789);
		System.out.println("Connected to localhost at port 6789");
		PrintWriter toserver = new PrintWriter(skt.getOutputStream(), true);
		BufferedReader fromserver = new BufferedReader(new
InputStreamReader(skt.getInputStream()));

		System.out.println("Enter an integer : ");
		String n = fromuser.readLine();
		toserver.println(n);
		skt.close();*/

		BufferedReader fromuser = new BufferedReader(new InputStreamReader(System.in));
		System.out.println("How Many Number You Want To Enter : ");

		int no = Integer.parseInt(fromuser.readLine());
		int[] dataword = new int[no];
		int sum = 0;
		for(int i=0; i<dataword.length; i++){
			System.out.print("Enter no : " + (i + 1) + " : ");
			dataword[i] = Integer.parseInt(fromuser.readLine());
			//System.out.println("Dataword ["+ (i+1) +"] is : " + dataword[i]);
			sum = sum + dataword[i];
		}
		System.out.println("Sum : " + sum);
		String dtbin = "" + Integer.toBinaryString(sum);

		//String dtbin = "110011101011011011";

		System.out.println("Sum In Binary : " + dtbin);

		int div_ans = dtbin.length()/4;

		int div_rem = dtbin.length()%4;

		System.out.println("Sum In Split : " + div_ans + " : " + div_ans);
		System.out.println("Rem : " + div_ans + " : " + div_rem);
		String[] str;
		int lenstr;
		if(div_ans != 0){
			str = new String[div_ans + 1];
			lenstr = str.length - 1;
		}
```

```
		else{
			str = new String[div_ans];
			lenstr = str.length - 1;
		}


		/*String str = "10111010";

		System.out.println("B1 : " + str.substring(4, 8));
		System.out.println("B2 : " + str.substring(0, 4));*/

		for(int st=dtbin.length()-4, end = dtbin.length(), i = 0; i<lenstr ; st = st-4, end = end - 4,
i++){
			//System.out.println("st : " + st + " : end : " + end);
			str[i] = dtbin.substring(st, end);
			System.out.println("str ["+i+"] : " + str[i]);
		}

		String tmp = "";
		if(div_rem != 0){
			for(int i=0; i<(4-div_rem); i++){
			tmp = tmp + "0";
			}

			str[div_ans] = tmp + dtbin.substring(0, (4-tmp.length()));
			//System.out.println("XXX : " + div_rem);
		}

		System.out.println("LEN : " + lenstr);

		for(int i=0; i<str.length; i++){
			System.out.println("A ["+(i)+"] : " + str[i]);
		}
		String d1="", d2="", ans=str[0];

		for(int i=0; i<str.length-1; i++){
			d1 = ans;
			d2 = str[i+1];
			ans = checksum(d1, d2);
			//ans = checksum("1011", "1101");
			System.out.println("1check sum ans : " + d1 + " : " + d2 + " : " + ans);
		}
		System.out.println("check sum ans : " + ans);
		char[] tmp1 = ans.toCharArray();
		String cs = "";
		for(int i=0; i<tmp1.length; i++){
			if(tmp1[i] == '0'){
				tmp1[i] = '1';
```

```
                }
                else{
                        tmp1[i] = '0';
                }
                cs = cs + tmp1[i];
                System.out.println("R : " + tmp1[i]);
        }
        System.out.println("Final check sum ans : " + cs);
}

public static String checksum(String s1, String s2){
        char c1='0', c2='0', c3 = '0', ans1='0';
        String ans="";
        for(int i=s1.length()-1; i>-1; i--){
                c1 = s1.charAt(i);
                c2 = s2.charAt(i);
                System.out.println("C3 : " + c3 + " : C1 : " + c1 + " : C2 : " + c2);

                if(c3 == '0' && c1 == '0' && c2 == '0'){
                        ans1 = '0';
                        c3 = '0';
                        //System.out.println("000 i : " + i);
                        }
                        else if(c3 == '0' && c1 == '0' && c2 == '1'){
                                ans1 = '1';
                                c3 = '0';
                                //System.out.println("001 i : " + i);
                        }
                        else if(c3 == '0' && c1 == '1' && c2 == '0'){
                                ans1 = '1';
                                c3 = '0';
                                //System.out.println("010 i : " + i);
                        }
                        else if(c3 == '0' && c1 == '1' && c2 == '1'){
                                ans1 = '0';
                                c3 = '1';
                                //System.out.println("011 i : " + i);
                        }
                        else if(c3 == '1' && c1 == '0' && c2 == '0'){
                                ans1 = '1';
                                c3 = '0';
                                //System.out.println("100 i : " + i);
                        }
                        else if(c3 == '1' && c1 == '0' && c2 == '1'){
                                ans1 = '0';
                                c3 = '1';
                                //System.out.println("101 i : " + i);
                        }
```

```
                        else if(c3 == '1' && c1 == '1' && c2 == '0'){
                                ans1 = '0';
                                c3 = '1';
                                //System.out.println("110 i : " + i);
                        }
                        else if(c3 == '1' && c1 == '1' && c2 == '1'){
                                ans1 = '1';
                                c3 = '1';
                                //System.out.println("111 i : " + i);
                        }
                        ans = ans1 + ans;
                }
                if(c3 == '1'){
                ans = checksum(ans, "0001");
        }
        System.out.println("ANS1 : " + ans1);
        System.out.println("ANS : " + ans);
        return ans;
    }
}
```

Hamming Code Sender :

```
import java.io.*;
import java.net.*;


public class HammingCodeSender {
    public static void main(String[] args)throws Exception {
        Socket s=new Socket("localhost",3333);
        DataInputStream din=new DataInputStream(s.getInputStream());
        DataOutputStream dout=new DataOutputStream(s.getOutputStream());
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

        int no = 1234;
        String dataword = Integer.toBinaryString(no);
        int pcnt = 1, k = 0;
        int noframe = 0, i = 0;
        String tmp = dataword;
        //Count No Of Parity Bit
        System.out.println("Data Word Is : " + dataword);
        while(true){
            if(tmp.isEmpty()){
                break;
            }
            if(Math.pow(2, k) == i){
                pcnt++;
                k++;
```

```
		}
		tmp = tmp.substring(0, tmp.length() - 1);
		i++;
	}
	System.out.println("No of Parity Bit require are : " + pcnt);
	//Create A Frame With Extra Parity Bit.
	char codeword[] = new char[(dataword.length() + pcnt) + 1];
	//Add '0' to make this codeword as per theory algorithm. (start from 1)
	codeword[0] = '0';
	k=0;
	for(int p=1, x=0; p<codeword.length; p++){
		if(Math.pow(2, k) != p){
			if(x == dataword.length()){
				break;
			}
			codeword[p] = dataword.charAt(x);
			x++;
		}
		else{
			k++;
			codeword[p] = '0';
		}
	}
	//Create And Calculate bit for Series of All Extra Parity Bit.(SP1, SP2, etc...)
	char paritybit[] = new char[pcnt+1];
	paritybit[0] = '0';
	int bit = (Integer.toBinaryString(codeword.length - 1)).length();
	String tmp1 = new String();
	String tmp2 = new String();
	for(int p=1; p<paritybit.length; p++){
		for(int x=0; x<=codeword.length-1; x++){
			int t = bit - (Integer.toBinaryString(x)).length();
			for(int m=0; m<t; m++){
				tmp1 = tmp1 + "0";
			}
			tmp1 = tmp1 + Integer.toBinaryString(x);
			if(tmp1.charAt(tmp1.length() - p) == '1'){
				tmp2 = tmp2 + codeword[x];
			}
			tmp1 = "";
		}
		//bits to calculating parity bit.
		System.out.println("SP" + p + " : " + tmp2);
		paritybit[p] = pbg(tmp2, "even");
		tmp2="";
	}
	//Set Values Into The ParityBit.
	System.out.println("Final CodeWord : ");
```

```
                for(int p=1, h=0, c=1; p<codeword.length; p++){
                        if(Math.pow(2, h) == p){
                                codeword[p] = paritybit[c];
                                h++;
                                c++;
                        }
                }
                System.out.println("Final : "+new String(codeword).substring(1));
                //Make Data With Error. Put '1' instead of '0'.

                if(codeword[5] == '1')
                        codeword[5] = '0';
                else
                        codeword[5] = '1';

                //Display Final Code Word Bit by Bit. Remove '0' that we added above.
                System.out.println("Codeword With Error at index [ 5 ] : " + new
String(codeword).substring(1));
                dout.writeUTF(new String(codeword).substring(1));

                dout.close();
                s.close();
        }

        public static char pbg(String x, String p){
                int cnt = 0;
                char pb;
                for(int i = 0; i < x.length(); i++){
                        if(x.charAt(i) == '1'){
                                cnt++;
                        }
                }
                if(p.equals("even")){
                        if(cnt % 2 == 0){
                                pb = '0';
                        }
                        else{
                                pb = '1';
                        }
                }
                else{
                        if(cnt % 2 == 0){
                                pb = '1';
                        }
                        else{
                                pb = '0';
                        }
                }
```

```
		return pb;
	}
}
```

Hamming Code Receiver :

```
import java.io.*;
import java.net.*;

public class HammingCodeReciver {
	public static void main(String[] args)throws Exception {
		ServerSocket ss=new ServerSocket(3333);
		Socket s=ss.accept();
		DataInputStream din=new DataInputStream(s.getInputStream());
		DataOutputStream dout=new DataOutputStream(s.getOutputStream());
		BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

		String ttt = din.readUTF();
		System.out.println("Code Word : " + ttt);
		//Add '0' to make this codeword as per theory algorithm. (start from 1)
		ttt = "0" + ttt;
		char codeword[] = ttt.toCharArray();
		int k = 0, pcnt = 0;
		//Count No Of Parity Bit In The CodeWord.
		for(int p=1; p<codeword.length; p++){
			if(Math.pow(2, k) == p){
				pcnt++;
				k++;
			}
		}
		System.out.println("No Of Parity Bit : " + pcnt);
		//Create And Calculate bit for Series of All Extra Parity Bit.(SP1, SP2, etc...)
		char paritybit[] = new char[pcnt+1];
		paritybit[0] = '0';
		int bit = (Integer.toBinaryString(codeword.length - 1)).length();
		String tmp1 = new String();
		String tmp2 = new String();
		for(int p=1; p<paritybit.length; p++){
			for(int x=0; x<=codeword.length-1; x++){
				int t = bit - (Integer.toBinaryString(x)).length();
				for(int m=0; m<t; m++){
					tmp1 = tmp1 + "0";
				}
				tmp1 = tmp1 + Integer.toBinaryString(x);
				if(tmp1.charAt(tmp1.length() - p) == '1'){
					tmp2 = tmp2 + codeword[x];
				}
```

```
                        tmp1 = "";
                    }
                    System.out.println("SP" + p + " : " + tmp2);
                    paritybit[p] = pbg(tmp2, "even");
                    tmp2="";
                }
                //Hamming Code
                String hammingcode = new String();
                for(int p=paritybit.length - 1, x=1; p>0; p--){
                    hammingcode = hammingcode + paritybit[p];
                }
                //Display Hamming Code.
                System.out.println("HammingCode : " + new String(hammingcode));
                //Find Decimal of Hamming Code
                char x[] = new String(paritybit).substring(1).toCharArray();
                int sum = 0;
                for(int p=0; p<x.length; p++){
                    if(x[p] == '1'){
                        sum = (int) (sum + Math.pow(2, p));
                    }
                }
                if(sum == 0){
                    System.out.println("Data Is Error Free.");
                }
                else{
                    System.out.println("Error Is At Location : " + sum);
                }
            din.close();
            s.close();
            ss.close();

}

public static char pbg(String x, String p){
                int cnt = 0;
                char pb;
                for(int i = 0; i < x.length(); i++){
                    if(x.charAt(i) == '1'){
                        cnt++;
                    }
                }
                if(p.equals("even")){
                    if(cnt % 2 == 0){
                        pb = '0';
                    }
                    else{
                        pb = '1';
                    }
```

```
                }
                else{
                        if(cnt % 2 == 0){
                                pb = '1';
                        }
                        else{
                                pb = '0';
                        }
                }
                return pb;
        }
}
```