Bansilal Ramnath Agarwal Charitable Trust's

Vishwakarma Institute of Technology, Pune-37

Department Of Artificial Intelligence and Data Science

COMPUTER NETWORK

Class: - SY BTECH

Branch: - AIDS

Prepared By: - Jineshwari Anil Bagul Year: -2024-25

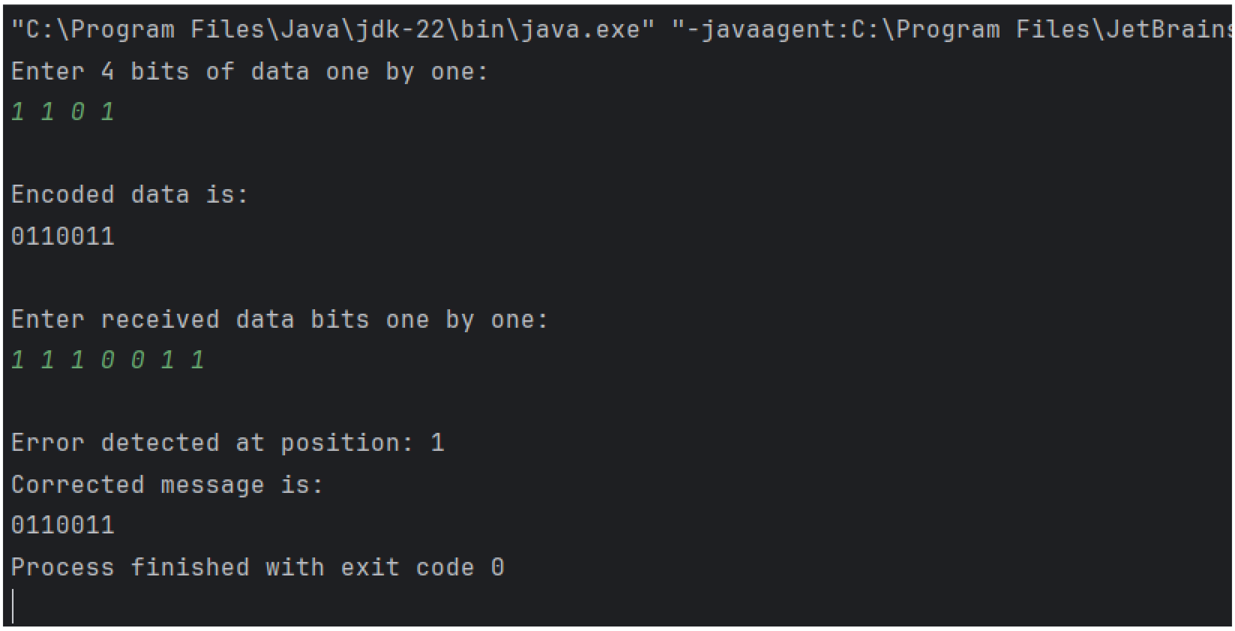
Assignment No - 5

TITLE: Write a program for error detection and correction for 7/8 bits ASCII codes using HammingCodes.

3 bit Parrity Hamming Code –

```
package CNLAB;

import java.util.Scanner;

public class Hamming_code {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int[] data = new int[8];
        int[] dataAtRec = new int[8];
        int c1, c2, c3, c;

        System.out.println("Enter 4 bits of data one by one:");
        data[7] = sc.nextInt();
        data[6] = sc.nextInt();
        data[5] = sc.nextInt();
        data[3] = sc.nextInt();

        // Calculation of even parity bits
        data[4] = data[5] ^ data[6] ^ data[7];
        data[2] = data[3] ^ data[6] ^ data[7];
        data[1] = data[3] ^ data[5] ^ data[7];

        System.out.println("\nEncoded data is:");
        for (int i = 1; i <= 7; i++) {
            System.out.print(data[i]);
        }

        System.out.println("\n\nEnter received data bits one by one:");
        for (int i = 1; i <= 7; i++) {
            dataAtRec[i] = sc.nextInt();
        }

        // Error detection
        c1 = dataAtRec[1] ^ dataAtRec[3] ^ dataAtRec[5] ^ dataAtRec[7];
        c2 = dataAtRec[2] ^ dataAtRec[3] ^ dataAtRec[6] ^ dataAtRec[7];
        c3 = dataAtRec[4] ^ dataAtRec[5] ^ dataAtRec[6] ^ dataAtRec[7];

        c = c3 * 4 + c2 * 2 + c1;
        if (c == 0) {
            System.out.println("\nCongratulations! There is no error.");
        } else {
            System.out.println("\nError detected at position: " + c);
            System.out.println("Corrected message is:");

            // Correct the error
            dataAtRec[c] = (dataAtRec[c] == 0) ? 1 : 0;

            for (int i = 1; i <= 7; i++) {
                System.out.print(dataAtRec[i]);
            }
        }

        sc.close();
    }
}
```

output-

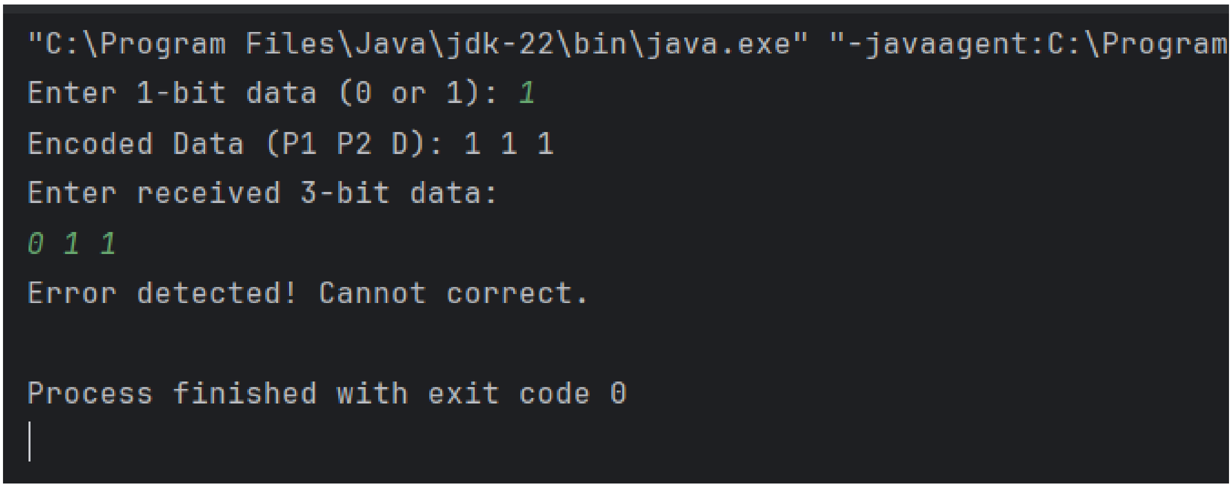


```
"C:\Program Files\Java\jdk-22\bin\java.exe" "-javaagent:C:\Program Files\JetBrains
Enter 4 bits of data one by one:
1 1 0 1

Encoded data is:
0110011

Enter received data bits one by one:
1 1 1 0 0 1 1

Error detected at position: 1
Corrected message is:
0110011
Process finished with exit code 0
```

2 bit Parrity Hamming Code –

```
package CNLAB;

import java.util.Scanner;

public class Hamming_code {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int[] data = new int[8];
        int[] dataAtRec = new int[8];
        int c1, c2, c3, c;

        System.out.println("Enter 4 bits of data one by one:");
        data[7] = sc.nextInt();
        data[6] = sc.nextInt();
        data[5] = sc.nextInt();
        data[3] = sc.nextInt();

        // Calculation of even parity bits
        data[4] = data[5] ^ data[6] ^ data[7];
        data[2] = data[3] ^ data[6] ^ data[7];
        data[1] = data[3] ^ data[5] ^ data[7];

        System.out.println("\nEncoded data is:");
        for (int i = 1; i <= 7; i++) {
            System.out.print(data[i]);
        }

        System.out.println("\n\nEnter received data bits one by one:");
        for (int i = 1; i <= 7; i++) {
            dataAtRec[i] = sc.nextInt();
        }

        // Error detection
        c1 = dataAtRec[1] ^ dataAtRec[3] ^ dataAtRec[5] ^ dataAtRec[7];
        c2 = dataAtRec[2] ^ dataAtRec[3] ^ dataAtRec[6] ^ dataAtRec[7];
        c3 = dataAtRec[4] ^ dataAtRec[5] ^ dataAtRec[6] ^ dataAtRec[7];

        c = c3 * 4 + c2 * 2 + c1;

        if (c == 0) {
            System.out.println("\nCongratulations! There is no error.");
        } else {
            System.out.println("\nError detected at position: " + c);
            System.out.println("Corrected message is:");

            // Correct the error
            dataAtRec[c] = (dataAtRec[c] == 0) ? 1 : 0;

            for (int i = 1; i <= 7; i++) {
                System.out.print(dataAtRec[i]);
            }
        }

        sc.close();
    }
}
```

OutPut-

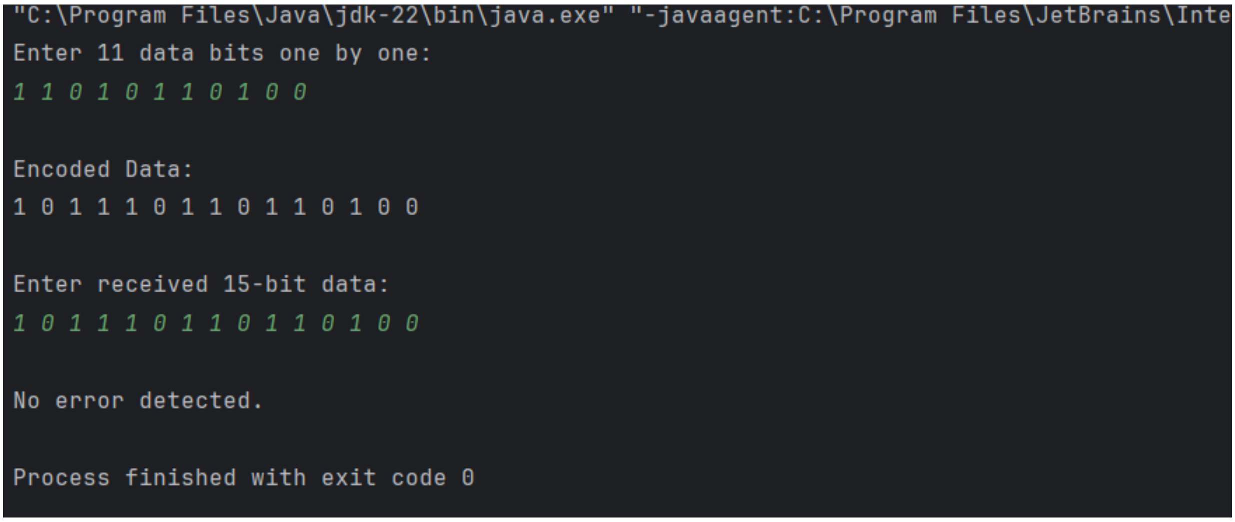


```
"C:\Program Files\Java\jdk-22\bin\java.exe" "-javaagent:C:\Program
Enter 1-bit data (0 or 1): 1
Encoded Data (P1 P2 D): 1 1 1
Enter received 3-bit data:
0 1 1
Error detected! Cannot correct.

Process finished with exit code 0
```

4 bit Parrity Hamming Code-

```
package CNLAB;

import java.util.Scanner;

public class Parity4 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int[] data = new int[16]; // 1-based indexing

        // Input 11 data bits
        System.out.println("Enter 11 data bits one by one:");
        int[] positions = {3, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15};
        for (int pos : positions) {
            data[pos] = sc.nextInt();
        }

        // Calculate 4 parity bits (Even Parity)
        data[1] = data[3] ^ data[5] ^ data[7] ^ data[9] ^ data[11] ^ data[13] ^ data[15];
        data[2] = data[3] ^ data[6] ^ data[7] ^ data[10] ^ data[11] ^ data[14] ^ data[15];
        data[4] = data[5] ^ data[6] ^ data[7] ^ data[12] ^ data[13] ^ data[14] ^ data[15];
        data[8] = data[9] ^ data[10] ^ data[11] ^ data[12] ^ data[13] ^ data[14] ^ data[15];

        // Output Encoded Data
        System.out.println("\nEncoded Data:");
        for (int i = 1; i <= 15; i++) {
            System.out.print(data[i] + " ");
        }
        System.out.println();

        // Input Received Data
        int[] received = new int[16];
        System.out.println("\nEnter received 15-bit data:");
        for (int i = 1; i <= 15; i++) {
            received[i] = sc.nextInt();
        }

        // Error Detection
        int c1 = received[1] ^ received[3] ^ received[5] ^ received[7] ^ received[9] ^ received[11]
^ received[13] ^ received[15];
        int c2 = received[2] ^ received[3] ^ received[6] ^ received[7] ^ received[10] ^ received[11]
^ received[14] ^ received[15];
        int c3 = received[4] ^ received[5] ^ received[6] ^ received[7] ^ received[12] ^
received[13] ^ received[14] ^ received[15];
        int c4 = received[8] ^ received[9] ^ received[10] ^ received[11] ^ received[12] ^
received[13] ^ received[14] ^ received[15];

        int errorPos = c4 * 8 + c3 * 4 + c2 * 2 + c1;

        if (errorPos == 0) {
            System.out.println("\nNo error detected.");
        } else {
            int a = 11-errorPos;
            System.out.println("\nError detected at position: " + a);
            received[errorPos] ^= 1; // Flip the incorrect bit
            System.out.println("Corrected Data:");
            for (int i = 1; i <= 15; i++) {
                System.out.print(received[i] + " ");
            }
        }

        sc.close();
    }
}
```

OutPut :-



```
"C:\Program Files\Java\jdk-22\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\Inte
Enter 11 data bits one by one:
1 1 0 1 0 1 1 0 1 0 0

Encoded Data:
1 0 1 1 1 0 1 1 0 1 1 0 1 0 0

Enter received 15-bit data:
1 0 1 1 1 0 1 1 0 1 1 0 1 0 0

No error detected.

Process finished with exit code 0
```