# PROGRAM10:

import java.util.Scanner; public class HammingCode {

public static void main(String[] args) { Scanner sc = new Scanner(System.in);

char[] data = new char[8]; // 1-based indexing char[] codeword = new char[13]; // 1 to 11 used

char[] received = new char[13]; // for received codeword System.out.print("Enter 7-bit data: ");

String input = sc.next(); if (input.length() != 7) {

System.out.println("Error: You must enter exactly 7 bits."); return;

}

for (int i = 0; i < 7; i++) { data[i + 1] = input.charAt(i);

}

int j = 0, k = 1;

for (int i = 1; i <= 11; i++) { if (i == Math.pow(2, j)) {

codeword[i] = '0'; // parity placeholder j++;

} else {

codeword[i] = data[k++];

}

}

for (int i = 0; i < 4; i++) {

int parityPos = (int) Math.pow(2, i); int count = 0;

for (int p = parityPos; p <= 11; p += 2 \* parityPos) {

for (int q = 0; q < parityPos && (p + q) <= 11; q++) { if (codeword[p + q] == '1') {

count++;

}

}

}

codeword[parityPos] = (count % 2 == 0) ? '0' : '1';

}

System.out.print("Generated codeword: "); for (int i = 1; i <= 11; i++) {

System.out.print(codeword[i]);

}

System.out.print("\n\nEnter received 11-bit Hamming code: "); String recv = sc.next();

if (recv.length() != 11) {

System.out.println("Error: You must enter exactly 11 bits."); return;

}

for (int i = 1; i <= 11; i++) { received[i] = recv.charAt(i - 1);

}

int errorPos = 0;

for (int i = 0; i < 4; i++) {

int parityPos = (int) Math.pow(2, i);

int count = 0;

for (int p = parityPos; p <= 11; p += 2 \* parityPos) {

for (int q = 0; q < parityPos && (p + q) <= 11; q++) { if (received[p + q] == '1') {

count++;

}

}

}

if (count % 2 != 0) { errorPos += parityPos;

}

}

if (errorPos == 0) {

System.out.println("\nNo error detected in received codeword.");

} else {

System.out.println("\nError detected at position: " + errorPos);

// Correct the error

received[errorPos] = (received[errorPos] == '1') ? '0' : '1'; System.out.print("Corrected codeword: ");

for (int i = 1; i <= 11; i++) { System.out.print(received[i]);

}

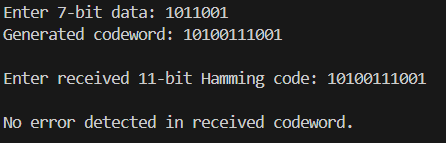
}

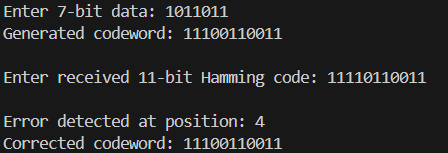
System.out.println(); sc.close();

}

}

**OUTPUT:**

****

****