**34. Implementing the simulation of error correction code - CRC in java/C.**

**#include <stdio.h>**

**#include <string.h>**

**#define POLYNOMIAL "1101" // Example divisor (CRC-3)**

**void xorOperation(char \*dividend, const char \*divisor) {**

**for (int i = 0; i < strlen(divisor); i++) {**

**dividend[i] = (dividend[i] == divisor[i]) ? '0' : '1';**

**}**

**}**

**void computeCRC(char \*data, char \*remainder, const char \*polynomial) {**

**int dataLen = strlen(data);**

**int polyLen = strlen(polynomial);**

**char temp[100]; // Ensure enough space**

**strcpy(temp, data);**

**for (int i = 0; i <= dataLen - polyLen; i++) {**

**if (temp[i] == '1') {**

**xorOperation(&temp[i], polynomial);**

**}**

**}**

**strncpy(remainder, &temp[dataLen - polyLen + 1], polyLen - 1);**

**remainder[polyLen - 1] = '\0';**

**}**

**void appendCRC(char \*data, const char \*remainder) {**

**strcat(data, remainder);**

**}**

**int verifyCRC(char \*receivedData, const char \*polynomial) {**

**char remainder[strlen(polynomial)];**

**computeCRC(receivedData, remainder, polynomial);**

**for (int i = 0; i < strlen(remainder); i++) {**

**if (remainder[i] != '0') return 0; // Error detected**

**}**

**return 1; // No error**

**}**

**int main() {**

**char data[100] = "11010011101100"; // Example data**

**char remainder[strlen(POLYNOMIAL)];**

**char transmittedData[100];**

**strcat(data, "000"); // Append space for remainder (POLYNOMIAL length - 1)**

**computeCRC(data, remainder, POLYNOMIAL);**

**strcpy(transmittedData, data);**

**appendCRC(transmittedData, remainder);**

**printf("Transmitted Data: %s\n", transmittedData);**

**// Simulate received data (introduce errors for testing)**

**char receivedData[100];**

**strcpy(receivedData, transmittedData);**

**receivedData[5] = (receivedData[5] == '0') ? '1' : '0'; // Introduce error**

**printf("Received Data: %s\n", receivedData);**

**if (verifyCRC(receivedData, POLYNOMIAL)) {**

**printf("No error detected.\n");**

**} else {**

**printf("Error detected.\n");**

**}**

**return 0;**

**}**

**A screenshot of a computer

AI-generated content may be incorrect.**