

B \$CRC Generator-

1. CRC stands for Cyclic Redundancy Check.
2. Cyclic Redundancy Check is a method of detecting accidental changes/errors in the communication channel.
3. It uses checksum in a digital network and storage device to detect error.

How CRC works:-

Receiver Side:-

1. Input data is in binary form.
2. Data is divided by input generator using modulo 2.
3. Remainder of the division is the checksum.
4. Remainder is added in the given data to form a codeword that is sent by the sender.

Sender Side:-

1. Receiver will receive the data with checksum.
2. Received data is then divided with a generator given by the user.
3. If the remainder of the division is zero
 - a. Then the received data is free from error.
4. If we don't get remainder is not zero
 - a. Then the received data has an error.
 - b. Error will be shown to the receiver.

Code Input-

```
#include <iostream>
#include <algorithm>
using namespace std;

void divide(int dividend[], int divisor[], int dataSize, int divisorSize,
int remainder[]) {
    for (int i = 0; i <= dataSize - divisorSize; ++i) {
        if (dividend[i] == 1) {
            for (int j = 0; j < divisorSize; ++j) {
                dividend[i + j] ^= divisor[j];
            }
        }
    }
    for (int i = 0; i < divisorSize - 1; i++) {
        remainder[i] = dividend[dataSize + i];
    }
}

void printArray(int arr[], int size) {
    for (int i = 0; i < size; i++) {
        cout << arr[i];
    }
}
```

```

    }
    cout << endl;
}

int main() {
    int data[5], generator[3], transmittedData[8], remainder[2];
    cout << "Enter 5-bit data: ";
    for (int i = 0; i < 5; i++) cin >> data[i];

    cout << "Enter 3-bit CRC generator: ";
    for (int i = 0; i < 3; i++) {
        cin >> generator[i];
    }

    for (int i = 0; i < 5; i++) {
        transmittedData[i] = data[i];
    }
    for (int i = 5; i < 7; i++) {
        transmittedData[i] = 0;
    }
    divide(transmittedData, generator, 5, 3, remainder);
    cout<<"(n-1) bits :2"<<endl;
    cout << "Remainder (extra bits): ";
    printArray(remainder, 2);

    int codeword[7];
    for (int i = 0; i < 5; i++) {
        codeword[i] = data[i];
    }
    for (int i = 0; i < 2; i++) {
        codeword[i + 5] = remainder[i];
    }
    cout << "Codeword: ";
    printArray(codeword, 7);
    cout << "\nReceiver---" << endl;
    int received[7];
    for (int i = 0; i < 7; i++) received[i] = codeword[i];
    int check[2];
    divide(received, generator, 5, 3, check);

```

```

    if (count(check, check + 2, 1) == 0) {
        cout << "No error detected!" << endl;
    } else {
        cout << "Error detected!" << endl;
    }
    char choice;
    cout << "Do you want to introduce an error? (y/n): ";
    cin >> choice;
    if (choice == 'y' || choice == 'Y') {
        int pos;
        cout << "Enter position to change (0-based index): ";
        cin >> pos;
        if (pos >= 0 && pos < 7) {
            codeword[pos] ^= 1;
        }
        cout << "Received Codeword with error: ";
        printArray(codeword, 7);
        cout<<"Error detected";
    }
    return 0;
}

```

Code Output-

```

PS C:\upes\sem4\DCN_LAB> ./main.exe
Enter 5-bit data: 1 1 0 1 1
Enter 3-bit CRC generator: 1 1 0
(n-1) bits :2
Remainder (extra bits): 00
Codeword: 1101100

Receiver---
No error detected!
Do you want to introduce an error? (y/n): y
Enter position to change (0-based index): 1
Received Codeword with error: 1001100
Error detected

```