**CRC**

**Code:**

#include <stdio.h>

#include <stdint.h>

#include <stdlib.h>

#define CRC8\_POLY 0x1D

uint8\_t calculate\_crc8(uint8\_t \*data, size\_t length)

{

    uint8\_t crc = 0x00;

    for (size\_t i = 0; i < length; i++)

    {

        crc ^= data[i];  // XOR the input byte with the current CRC value

        // Process each bit in the byte

        for (uint8\_t j = 8; j > 0; j--)

        {

            if (crc & 0x80)   // If the MSB is set

                crc = (crc << 1) ^ CRC8\_POLY;  // Shift left and XOR with the polynomial

            else

                crc <<= 1;  // Just shift left if MSB is not set

        }

    }

    return crc;

}

void append\_crc(uint8\_t \*data, size\_t length)

{

    uint8\_t crc = calculate\_crc8(data, length);

    data[length] = crc;  // Append CRC to the data (CRC-8 codeword)

}

int verify\_crc(uint8\_t \*data, size\_t length)

{

    uint8\_t received\_crc = data[length - 1];

    data[length - 1] = 0;

    // Calculate CRC over the received data (without the CRC byte)

    uint8\_t calculated\_crc = calculate\_crc8(data, length - 1);

    return (calculated\_crc == received\_crc);  // Return 1 if CRC matches, else 0

}

void print\_hex(uint8\_t \*data, size\_t length)

{

    for (size\_t i = 0; i < length; i++)

        printf("%02X ", data[i]);

    printf("\n");

}

int main()

{

    size\_t data\_length;

    printf("Sender Side:\n");

    printf("Enter the length of the data: ");

    scanf("%zu", &data\_length);

    uint8\_t \*data = (uint8\_t \*)malloc(data\_length \* sizeof(uint8\_t));

    if (data == NULL)

    {

        printf("Memory allocation failed!\n");

        return 1;

    }

    printf("Enter the data (hexadecimal values separated by spaces):\n");

    for (size\_t i = 0; i < data\_length; i++)

        scanf("%hhx", &data[i]);

    // Append CRC to the data

    append\_crc(data, data\_length);

    printf("Data with CRC-8 codeword (sent by sender): ");

    print\_hex(data, data\_length + 1);  // +1 for the CRC byte

    printf("\nReceiver Side:\n");

    printf("Enter the length of the received data (including CRC): ");

    size\_t received\_data\_length;

    scanf("%zu", &received\_data\_length);

    uint8\_t \*received\_data = (uint8\_t \*)malloc(received\_data\_length \* sizeof(uint8\_t));

    if (received\_data == NULL)

    {

        printf("Memory allocation failed!\n");

        free(data);

        return 1;

    }

    printf("Enter the received data (hexadecimal values separated by spaces):\n");

    for (size\_t i = 0; i < received\_data\_length; i++)

        scanf("%hhx", &received\_data[i]);

    if (verify\_crc(received\_data, received\_data\_length))

        printf("CRC verification succeeded!\n");

    else

        printf("CRC verification failed!\n");

    printf("Received Data: ");

    print\_hex(received\_data, received\_data\_length);

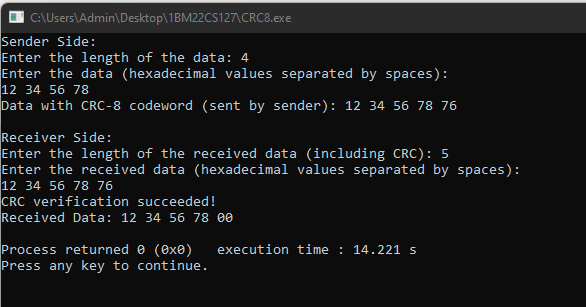
    free(data);

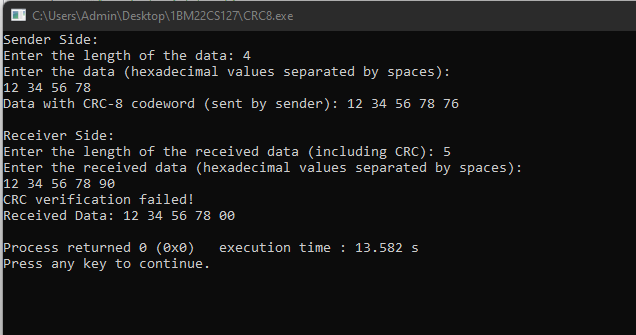
    free(received\_data);

    return 0;

}

**Output:**

****

****