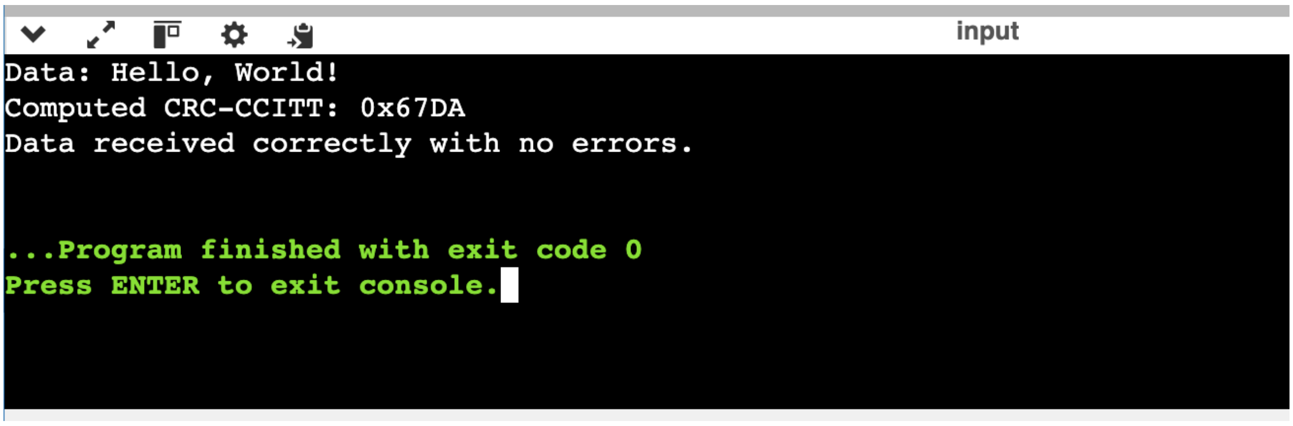
Error detection code using crc ccitt

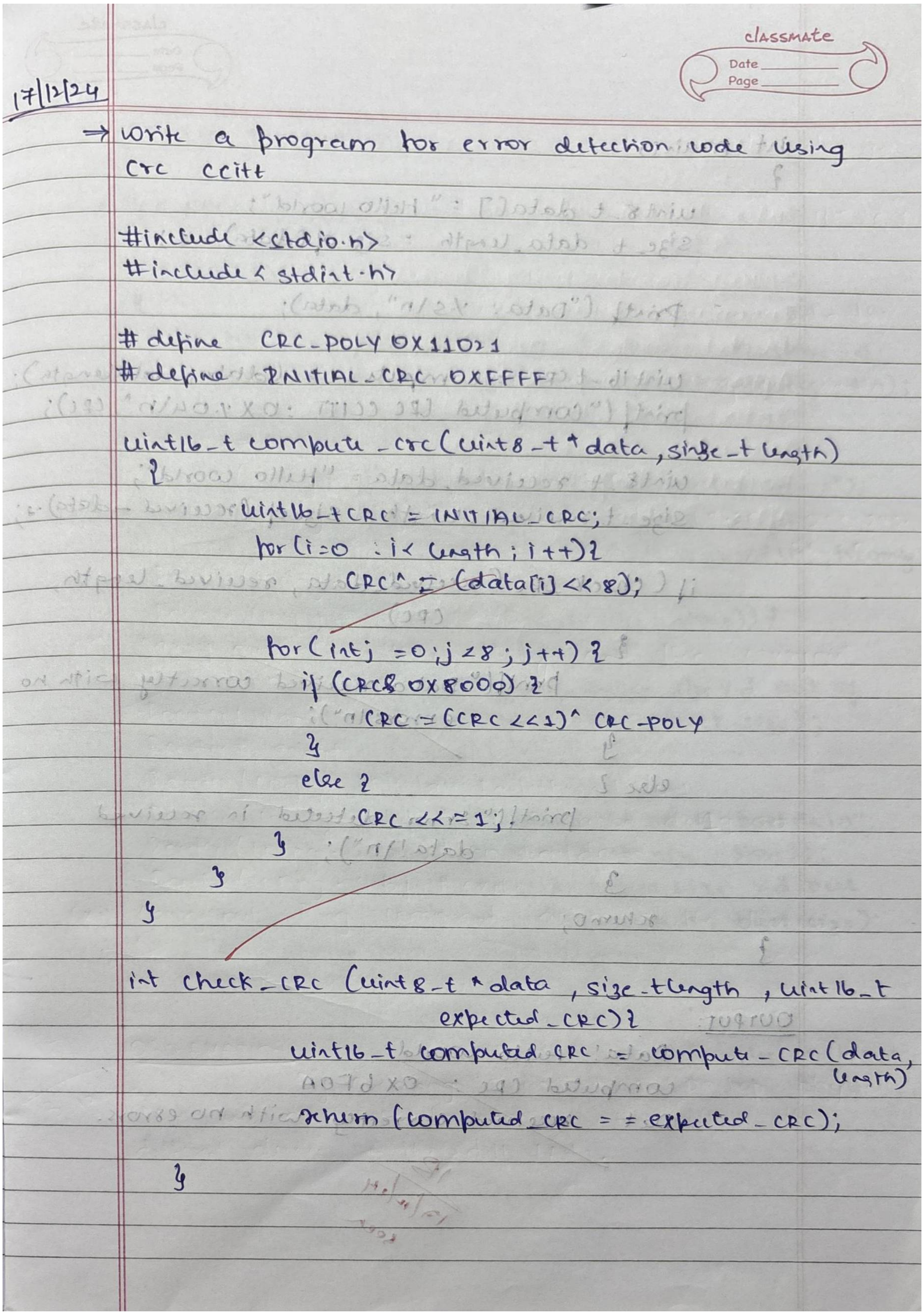
```
#include <stdio.h>
#include <stdint.h>

#define CRC_POLY 0x11021   // Polynomial for CRC-CCITT (x^16 + x^12 + x^5 + 1)
#define INITIAL_CRC 0xFFFF // Initial value of CRC for CRC-CCITT

// Function to compute the CRC for a given data buffer
uint16_t compute_crc(uint8_t *data, size_t length) {
    uint16_t crc = INITIAL_CRC;

    // Process each byte in the input data
    for (size_t i = 0; i < length; i++) {
        crc ^= (data[i] << 8); // Move byte into upper byte of CRC

        // Process each bit of the byte
        for (int j = 0; j < 8; j++) {
            if (crc & 0x8000) {
                crc = (crc << 1) ^ CRC_POLY; // Shift left and apply the polynomial if the leftmost bit is 1
            } else {
                crc <<= 1; // Otherwise just shift left
            }
        }
    }

    return crc & 0xFFFF; // Return the CRC (ensure it's 16 bits)
}

// Function to check if the received data is valid
int check_crc(uint8_t *data, size_t length, uint16_t expected_crc) {
    uint16_t computed_crc = compute_crc(data, length);
    return (computed_crc == expected_crc); // Return 1 if CRC matches, else 0
}

// Main function to demonstrate CRC-CCITT
int main() {
    uint8_t data[] = "Hello, World!"; // Example data
    size_t data_length = sizeof(data) - 1; // Exclude the null terminator

    printf("Data: %s\n", data);

    // Compute the CRC for the data
    uint16_t crc = compute_crc(data, data_length);

    // Display the computed CRC
    printf("Computed CRC-CCITT: 0x%04X\n", crc);

    // Simulate a transmission and check for errors by comparing CRC
    uint8_t received_data[] = "Hello, World!";
    size_t received_length = sizeof(received_data) - 1;

    // Check if the received data has the correct CRC
    if (check_crc(received_data, received_length, crc)) {
        printf("Data received correctly with no errors.\n");
    } else {
        printf("Error detected in received data!\n");
    }

    return 0;
}
```

Output:



```
Data: Hello, World!
Computed CRC-CCITT: 0x67DA
Data received correctly with no errors.


...Program finished with exit code 0
Press ENTER to exit console.
```

Observation:



classmate
Date
Page

17/12/24

→ Write a program for error detection code using crc ccitt

```
#include <stdio.h>
#include < stdint.h>

# define   CRC_POLY 0X11021
# define   INITIAL_CRC  0XFFFF

uint16_t compute_crc(uint8_t * data, size_t length)
  {
        uint16_t CRC = INITIAL_CRC;
            for (i=0   ; i< length ; i++){
                  CRC^ = (data[i] << 8);

             for( int j =0 ; j <8 ; j++) {
                if (CRC& 0X8000) {
                  CRC = (CRC <<1)^ CRC_POLY
                }
                else {
                    CRC <<=1;
                }
          }
  }

int check_CRC (uint8_t * data , size_t length , uint16_t
                           expected_CRC) {
           uint16_t computed CRC = compute_CRC(data,
                                                   length)
           return (computed_CRC = = expected_CRC);

   }
```

```
int main()
{
        uint8_t data[] = "Hello world";
        size_t data_length = sizeof(data) - 1);

        Printf ("Data: %s\n", data);

        uint16_t CRC = compute_CRC(data, datalength)
        printf("Computed CRC CCITT : 0X%04\n", CRC);

        uint8_t received_data = "Hello world";
        size_t received_length = sizeof (received_data)-1

        if (check_CRC (received_data, received_length,
                              CRC)
            {
                printf("Data received correctly with no
                              errors\n");
            }
        else {
                printf("Error detected in received
                              data!\n");
            }
        return 0;
}
```

OUTPUT:

```
Data! Hello world
computed CRC : 0X6F0A
Dat received correctly with no errors.
```

13
12/11/24
Geeh