

## Program 1

**Aim of the program:** Write a program for error detecting code using CRC-CCITT (16-bits).

**Code:**

```
#include <stdio.h>
#include <stdint.h>
#define CRC_POLY 0x1021
#define INITIAL_CRC 0xFFFF
uint16_t compute_crc(uint8_t *data, size_t length) {
    uint16_t crc = INITIAL_CRC;
    for (size_t 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;
            }
        }
    }
    return crc & 0xFFFF;
}

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, data_length);
    printf("Computed CRC-CCITT: 0x%04X\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;
}
```

```

1 #include <stdio.h>
2 #include <stdint.h>
3 #define CRC_POLY 0x1021
4 #define INITIAL_CRC 0xFFFF
5 uint16_t compute_crc(uint8_t *data, size_t length) {
6     uint16_t crc = INITIAL_CRC;
7     for (size_t i = 0; i < length; i++) {
8         crc ^= (data[i] << 8);
9         for (int j = 0; j < 8; j++) {
10             if (crc & 0x8000) {
11                 crc = (crc << 1) ^ CRC_POLY;
12             } else {
13                 crc <<= 1;
14             }
15         }
16     }
17     return crc & 0xFFFF;
18 }
19 int check_crc(uint8_t *data, size_t length, uint16_t expected_crc) {
20     uint16_t computed_crc = compute_crc(data, length);
21     return (computed_crc == expected_crc);
22 }
23 int main() {
24     uint8_t data[] = "Hello, World!";
25     size_t data_length = sizeof(data) - 1;
26     printf("Data: %s\n", data);
27     uint16_t crc = compute_crc(data, data_length);
28     printf("Computed CRC-CCITT: 0x%04X\n", crc);
29     uint8_t received_data[] = "Hello, World!";
30     size_t received_length = sizeof(received_data) - 1;
31     if (check_crc(received_data, received_length, crc)) {
32         printf("Data received correctly with no errors.\n");
33     } else {
34         printf("Error detected in received data!\n");
35     }
36     return 0;
37 }

```

1/2/24

write a program for error detecting code using CRC-CCITT  
(16 bit)

```
#include <stdio.h>
#include <stdint.h>
#define CRC_POLY 0x1021
#define INITIAL_CRC 0xFFFF

uint_t compute_crc(uint8_t *data, size_t len) {
    uint16_t crc = INITIAL_CRC;
    for (size_t i = 0; i < len; i++) {
        crc ^= (data[i] << 8);
        for (int j = 0; j < 8; j++) {
            if (crc & 0x8000) {
                crc = (crc << 1) ^ CRC_POLY;
            } else {
                crc << 1;
            }
        }
    }
    return crc & 0xFFFF;
}

int check_crc(uint8_t *data, size_t len,
              uint16_t expected_crc) {
    uint16_t computed_crc = compute_crc(
        data, len);
    return computed_crc == expected_crc;
}

int main() {
    uint8_t data[] = "Hello World!";
```

CLASSMATE  
Date \_\_\_\_\_  
Page \_\_\_\_\_

```

size_t data_length = sizeof(data) - 1;
printf("Data: %s\n", data);
uint16_t crc = compute_crc(data, data_length);
printf("Computed CRC-CCITT: 0x%04x\n", crc);
if (check_crc(received_data, received_data_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.

```

**Output:**

```

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

```

```

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

```