

Lab-9

9.1)C program for 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 (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;
}
```

output:=

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

OBSERVATION: =

② write a program for error detecting 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 (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[];
    size_t data_length = sizeof(data) - 1;
    printf("Data: %s\n", data);
    uint16_t crc = compute_crc(data, data_length);
    printf("CRC: %04X\n", crc);

    uint8_t received_data[] = "Hello, World!";
    size_t received_length = sizeof(received_data) - 1;
}
```

it checked_crc (received_data, received_length, crc) {
 if (res {
 printf ("Error\n");
 }
 return 0
}

Output →

Data: Hello, World!

Computed CRC-CCITT: 0X67DA

Data Received correctly with no errors

100%
18/10/24
Amr

9.2)C PROGRAM FOR LEAKY BUCKET

// Online C compiler to run C program online

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

```
int main(){
```

```
    int incoming, outgoing, buck_size, n, store = 0;
```

```
    printf("Enter bucket size, outgoing rate and no of inputs: ");
```

```
    scanf("%d %d %d", &buck_size, &outgoing, &n);
```

```
    while (n != 0) {
```

```
        printf("Enter the incoming packet size : ");
```

```
        scanf("%d", &incoming);
```

```
        printf("Incoming packet size %d\n", incoming);
```

```
        if (incoming <= (buck_size - store)){
```

```
            store += incoming;
```

```
            printf("Bucket buffer size %d out of %d\n", store, buck_size);
```

```
        } else {
```

```
            printf("Dropped %d no of packets\n", incoming - (buck_size - store));
```

```
            printf("Bucket buffer size %d out of %d\n", store, buck_size);
```

```
            store = buck_size;
```

```
        }
```

```
        store = store - outgoing;
```

```
        printf("After outgoing %d bytes left out of %d in buffer\n", store, buck_size);
```

```
        n--;
```

```
    }
```

output:=

Enter bucket size, outgoing rate and no of inputs: 4

3

3

Enter the incoming packet size : 5

Incoming packet size 5

Dropped 1 no of packets

Bucket buffer size 0 out of 4

After outgoing 1 bytes left out of 4 in buffer

Enter the incoming packet size : 5

Incoming packet size 5

Dropped 2 no of packets

Bucket buffer size 1 out of 4

After outgoing 1 bytes left out of 4 in buffer

Enter the incoming packet size : 7

Incoming packet size 7

Dropped 4 no of packets

Bucket buffer size 1 out of 4

After outgoing 1 bytes left out of 4 in buffer

=== Code Execution Successful ===

C program Leaky Bucket program →

17-12-2024

(20)

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

```
int main() {
```

```
    int incoming, outgoing, bucket_size, n, store = 0;
```

```
    printf ("no of Inp");
```

```
    scanf ("%d %d %d", &bucket_size, &outgoing, &n);
```

```
    while (n != 0) {
```

```
        printf ("Enter packet-size:");
```

```
        scanf ("%d", &incoming);
```

```
        printf ("incoming packet size %d\n", incoming);
```

```
        if (incoming <= bucket_size - store) {
```

```
            store += incoming;
```

```
            printf ("Bucket Buffer size %d out of %d\n", store, bucket_size);
```

```
        } else {
```

```
            printf ("Dropped %d no of packets\n", incoming - (bucket_size - store));
```

```
            printf ("Bucket Buffer size %d out of %d\n", store, bucket_size);
```

```
            store = bucket_size;
```

```
            store = store - outgoing;
```

```
            printf ("After outgoing %d bytes left out of %d\n", store, bucket_size);
```

```
            n--;
```

```
        }
```

Output =

Enter bucket size, outgoing rate and no of inputs : 4 3 3

Enter the incoming packet size :

Incoming packet size : 5

Drop 1 no of packets

Buffer size 0 out of 4

After outgoing 1 bytes left out of 4 in Buffer

Enter the incoming packet size :

incoming packet size :

Dropped 2 no of packets

Bucket Buffer size 1 out of 4

After outgoing 1 bytes left out of 4 in buffer

Leaky Bucket
17/12/24
8:44

Enter the incoming packet size 17
incoming packet size 7

Dropped 4 no of packets

Dropped 4 no of packets

Bucket Buffer size 1 out of 4

After outgoing 1 bytes left out 4 in Buffer