LAB WEEK 10

1. Write a program for error detecting code using CRC-CCITT (16 bits).

```
Code:
#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.

Figure 1: Output of CRC-CCIT

```
25
17/12/24
  Write a program for error detecting code using
  CRC-CCITT (16-bits)
 Code:
 Hinclude estdio.hz
 #include estdint.h>
# define CRC-POLY OX11021
Hdefine INITIAL_CRC OXFFFF
clintl6.t compute.crc (u'int8_t *data, size.t length) ?
    cimtl6_+ crc = INITIAL-CRC;
    for (size + i =0; i < length; i++){
      Crc = (data[:]ec8);
      for (int j=0; j <8; j+1) {
          it (cre f 0 x3000) {
             Crc = (coc<<1) ^ (RC-POLY;
          3 else {
             Crc Keel;
    return crcfoxFFFF;
3
int check-circ (wints + *data, size + length, wint lat expected circ) {
    uint16_t computed_crc = compute_crc(data, length);
    return (computed-crc == expected-crd);
2
int main () {
   wint8-t data [] = "Hello world";
    size-t data length = size of (data) - 1;
    printf ("Data: V.s\n", data);
    uint16_t crc = compute_coc (data, data-length);
   printf ("computed SRC- (CITT: OX'. OAX'n", COO);
    uint 8_t received data []="Hello, world!"
   Size t received_length = Size of (received_length, cre)) ?
       print ("Data received correctly with no errors, ");
   3
```

Figure 2: Observation Book 1

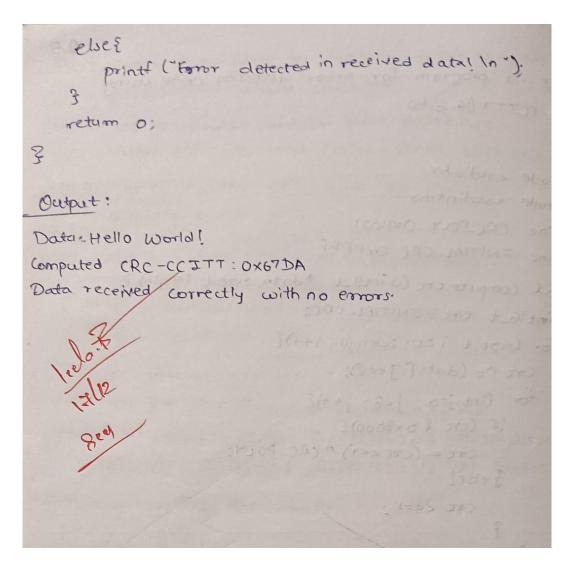


Figure 3: Observation Book 3

2. Write a program for congestion control using Leaky Bucket algorithm

Code:

```
#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: 10 3 3
Enter the incoming packet size : 5
Incoming packet size 5
Bucket buffer size 5 out of 10
After outgoing 2 bytes left out of 10 in buffer
Enter the incoming packet size : 5
Incoming packet size 5
Bucket buffer size 7 out of 10
After outgoing 4 bytes left out of 10 in buffer
Enter the incoming packet size : 7
Incoming packet size 7
Dropped 1 no of packets
Bucket buffer size 4 out of 10
After outgoing 7 bytes left out of 10 in buffer
```

Figure 4: Output for Leaky Bucket algorithm

```
17/12/24
 Write a program for congestion control using leaky
  Bucket algorithm
   Code
 #include <stdio.h>
 int main () {
     int incoming, outgoing, buck_size, n, store = 0;
      printf ('fater bucketsize, outgoing rate and no of inputs, ");
      Scanf ("1.d. 1.d", flouck-size, foutgoing, &n);
      while (n!=0) {
        print f( Finter the incoming packet size: );
        scanf (""d", lincoming);
         prints ("Incoming packet size vidla", incoming);
        if (incoming 5= (buck-size -store)){
            store +=incoming;
            printf ("Bucket buffor size 12d out of 12d/n", store.
         3 eve ?
           printf (" Dropped Y.d no of packets in "incoming -
                                      (buck_size - store);
           prints ("Bucket buffer size 1.d out of 7.d ) , store,
                                     buck size);
           Store = buck_size;
         Store = store - outgoing;
         prints ("After outgoing 1 d bytes left out of 1.d in
                buffer in , store, buck size);
       4
```

Figure 5: Observation Book 1

Ocetput inputs: Enter bucket size, outgoing rate and no of output 10 3 3 Enter the incoming packet size: 5 Incoming packet size 5 Bucket buffer size 5 out of 10 After outgoing 2 bytes left out of 10 in buffer Enter the incoming packet size :5 Incoming packet size: 5 Burcket buffer size ? out of 10 After outgoing 4 bytes left out of 10 in buffer Enter the incoming packet size: 7 Incoming packet size 7 Dropped I no of packets Bucket buffer size 4 out of 10 After outgoing 7 bytes left out at 10 in buffer

Figure 6: Observation Book 2