1. Write a program for error detecting code using CRC-CCITT.

```
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++) {
     \operatorname{crc} \triangleq (\operatorname{data}[i] \ll 8);
     for (int j = 0; j < 8; j++) {
       if (crc & 0x8000) {
          crc = (crc << 1) \land 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:

Output

```
Data: Hello, World!

Computed CRC-CCITT: 0x67DA

Data received correctly with no errors.
```

Observation:

```
Write a program for error detecting code using (RC-CCITT (16-bits).
 # include < stdio . h>
# include < stdint . h>
 # define CRC-POLY OXHOOL
 # define INTITAL_CRC OXFFFF
 Unit 16 t (ompute irc (vint 8 t * Jata, size t length)?

Vint 16 t (rc = TNITIAL - CRC;
  for (sig. t ?=0; {<length; i++){
        for (int j = 0 : j < 8; j++)?
            ?f (crc & 0x8000) {
               cre = (cress) 1 (RC_POLY;
             y else f
             y (recess);
   return crc + 0xFFF;
int check_crc(vints_t * data, size t length
               untle t expected crest
   vintes t computed cre = compute cre (data, length);
   setvin (computed_crc == expected_crc);
int main() }
  vint 8.t data [ = "Hello, World!";
   sizet date length = Size of (data) -1;
print f (" Pata: "/s)n", data);
   vintlet (rc = compute-cre(data, data-length):
   printf ("Computed (RC-CCTTT: 0x %04x10", creli
```

Violate recieved data [] = "Hella Dolld!";

Style te recieved length - Speof (recieved data) - 1;

if (check-crc (recieved data, recieved data) - 1;

printf ("Pata recieved correctly with No

elsef

printf ("Error detected in recieved data!\n");

return 0;

y

Data: Hello, World!

(computed CRC-CCTTT: 0x67DA

Data recieved correctly with no errors.

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);
  }
}
```

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
```

Observation:

```
Write a program for prior detecting code
    Using (RC-CCITT (16-bits).
  # include < stdio h>
  # include < stdint . h>
 # define (RC-POLY OXIIO2)
 # define INTITAL_CRC OXFFFF
 unit 16_t compute irc (vint 8_t > data, size t length) {
   Untlat cre = INITIAL - CRES
   for (size t ?=0; i < length: i++) {
       (rc 1 = (data [i) < 8);
       for (int j = 0 °, j x 8; j++) {
           ?f(crc & 0x8000) {
             ere = (creasi) ^ (RC_POLY;
           y else f
             Crc24-1;
      3
  reduin crc + 0xFFF;
     check_crc(vint8_t * data, size_t length
                uintle t expected crest
   vinte. t computed cre = compute cre (data, length);
  setvin (computed_cre == expedid_cre);
int main() {
  vint8.t data = "Hello World!";
  size t data length = size of (data) -1;
print f ("Pata: "s)n", data);
   vintlet (rc = compute-cre(data, data-length);
   printf ("Computed CRI-CITTI: 0x %04x10", (re);
```

Vints t recieved data [] = "Hello, Doild!";

Size t recieved length = Sizeof (recieved data)-1;

If (check-cic (recieved data, recieved length, crc));

Printf ("Pata recieved correctly with no error. In");

else [

printf ("Error detected in recieved data!In");

return 0;

return 0;

Data: Hello, World!

Computed (RC-CCTTT: OX 6 TDA)

Data recieved correctly with no errors.