Lab - 9 [i]

Write a Program for error detecting code using CRC

```
#include <stdio.h>
#include <string.h>
#define N strlen(Poly)
char data[30];
char check_value[30];
char Poly[10];
int data_length, i, j;
Void xor
{
  for (j=1; j<N; j++)
   check value [j] = (check_value [j] == poly [j])?0:1;
}
Void receiver()
{
  printf ("Enter the received data :");
  Scanf ("%s", data);
  printf ("Data received : %s", data);
  crc();
for (i=0; i<N-1 && (check_value[i] != '1'); i++)
  if (N-1)
   printf ("Error detected");
  else
   printf ("No error detected");
}
Void crc()
{
  for (i=0; i<N; i++)
   check value [i] = data [i];
do {
```

```
if (check value[0] == '1') {
    XOR();
for (j=0; j<N-1; j++)
    check_value[j] = check value[j+1]
    check_volue[j] = data[j+i];
} while (i <= data.length + N-1);

}
int main()
{
    printf ("Enter data to be transmitted: ");
    scanf ("%s", &data);
    printf (" Enter the divisor polynomial: ");
    Scanf ("%s", &poly);
    data.length = strelen (data);
    for (i = data.length ; i < datalength + N-1 ; i++)
        data[i]=0;
    printf ("Data padded with n-1 zeros : %s ) data
    crc();
    printf (("CRC value is %s") check_value);
    for (i= datalength ; i < data.length +N-1 ; i++)
        data[i] = check value [i-data.length];
    printf ("final dataword to be sent : %s" da
    receiver ();
    return 0;
}
```

Output:

Enter data to be transmitted : 101010

Enter the divisor polynomial : 1011

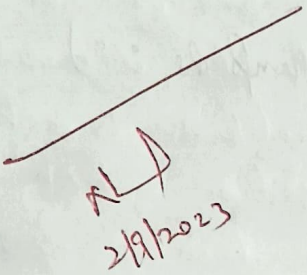
Data padded with zeros : 101010000

CRC value is : 001

Final codeword to be sent : 101010001

Enter the reciver data : 100010000

Error detected.



2/9/2023

Lab 6.09 [ii]

Write a program for Congestion Control using leaky bucket algorithm

```
#include <stdio.h>
int main()
{
 int incoming, outgoing, buck_size, n, store=0;
 printf("Enter the bucket size: ");
 scanf("%d", &outgoing);
 printf("Enter the no of inputs: ");
 scanf("%d", &n);
 while(n!=0)
 {
  printf("Enter the incoming bucket size: ");
  scanf("%d", &incoming);
  if(incoming <= (buck_size-store))
  {
   store = incoming;
   printf("Bucket Buffer size %d out of %d\n", s
          bucket size);
  }
 else
{
printf("Dropped %d no of packets\n");
 incoming = (bucket_size - store);
printf("Bucket Buffer Size %d of %d \n", store
          buck_size);
 Store = buck_size;
}
```

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

Output:

Enter bucket size = 5000

Enter outgoing rate = 2000

Enter number of ~~packets~~ inputs : 2

Enter the incoming packet size = 3000

Bucket Buffer size 3000 out of 5000

After outgoing 1000 packets left out of 5000 in buffer

Enter the incoming packet size : 1000

Bucket buffer size 2000 out of 5000

After outgoing 0 packets left out of 5000 in buffer.

NP
2/9/2023

```
Enter data to be transmitted: 1000100000100001

 Enter the Generating polynomial: 1011

 Data padded with n-1 zeros : 10001000001000001000
CRC or Check value is : 100
 Final data to be sent : 10001000001000001100
Enter the received data: 10001000001000001100
Data received: 10001000001000001100
No error detected
```

```
Enter data to be transmitted: 1000100000100001

 Enter the Generating polynomial: 1011

 Data padded with n-1 zeros : 10001000001000001000
CRC or Check value is : 100
 Final data to be sent : 10001000001000001100
Enter the received data: 10010000001000001100
Data received: 10010000001000001100
Error detected
```

```
Enter the bucket size:
5000
Enter the outgoing data rate:
200
Enter the size of incoming packet
3000
Data packet is accepted
Remaining space in bucket is.... 2200

Enter the size of incoming packet
2500
Data packet is dropped because the bucket size is less than the packet size

Enter the size of incoming packet
```