

## Cycle 2

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

classmate  
 Date \_\_\_\_\_  
 Page \_\_\_\_\_

Cycle 2

```

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

#include <stdio.h>
#include <string.h>

void binaryXOR (char *result, const char *a,
               const char *b) {
    for (int i=0; i<16; i++) {
        result[i] = (a[i] != b[i]) ? '0' : '1';
    }
    result[16] = '\0';
}

void CRC (const char *data, int length,
          char *checksum) {
    char crc[17];
    for (int i=0; i<16; i++) {
        crc[i] = '0';
    }
    crc[16] = '\0';
    for (int i=0; i<length; i++) {
        for (int j=0; j<8; j++) {
            char msb = crc[0];
            for (int k=0; k<16; k++) {
                crc[k] = crc[k+1];
            }
            crc[15] = '0';
            if (msb == '1') {
                char temp[17];
                binaryXOR (temp, crc,
                          "1000100000010001");
                strcpy (crc, temp);
            }
        }
    }
  
```

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

```

    }
    crc[15] = [data[i] == '1' ? '1' : '0'];
}
strcpy(checksum, crc);
}

void main() {
    char data[100];
    printf("Enter data in binary: ");
    scanf("%s", data);

    int datalength = strlen(data);
    char checksum[17];
    calculateCRC(data, datalength, checksum);

    char receivedChecksum[17];
    printf("Enter received CRC: ");
    scanf("%s", receivedChecksum);

    if (strcmp(receivedChecksum, checksum) == 0) {
        printf("Data is error-free\n");
    } else {
        printf("Data contains errors.\n");
    }

    return 0;
}

```

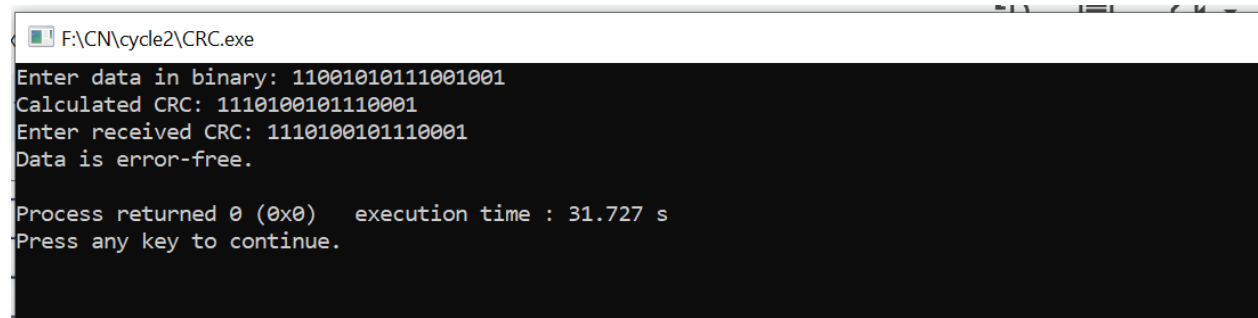
Output:

```

Enter data in binary: 11001010111001001
Calculated CRC: 1110100101110001
Entered received CRC: 1110100101110001
Data is error-free.

```

Output:



```
F:\CN\cycle2\CRC.exe
Enter data in binary: 11001010111001001
Calculated CRC: 1110100101110001
Enter received CRC: 1110100101110001
Data is error-free.

Process returned 0 (0x0)   execution time : 31.727 s
Press any key to continue.
```

14) Write a program for congestion control using Leaky bucket algorithm.

Write a program for congestion control using  
Leaky bucket algorithm

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

```
void main() {
```

```
    int psize, bsize, outgoing, emptySpace, choice;
```

```
    printf("Enter the bucket size : ");
```

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

```
    emptySpace = bsize;
```

```
    printf("Enter the outgoing rate : ");
```

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

```
    while(1) {
```

```
        printf("\nEnter the packet size: ");
```

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

```
        if (psize < bsize && psize <= emptySpace)
```

```
        {
```

```
            emptySpace = emptySpace - psize;
```

```
            printf("The packet of size %d is  
added and in the bucket \n", psize);
```

```
            emptySpace += outgoing;
```

```
        }
```

```
        else {
```

```
            printf("The packet of size %d is  
dropped due to lack of space in the  
bucket \n");
```

```
            printf("\nEnter 1 to continue or  
0 to stop: ");
```

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

```
            if (choice == 0)
```

```
                break;
```

```
        }
```

```
    }
```



Output:

Enter the bucket size : 5000

Enter the outgoing rate : 200

Enter the packet size : 3000

The packet of size 3000 is added and in the bucket

Enter 1 to continue or 0 to stop: 1

Enter the packet size : 2000

The packet of size 2000 is added and in the bucket

Enter 1 to continue or 0 to stop: 1

Enter the packet size : 1500

The packet of size 15000 is dropped due to lack of space in the bucket.

Enter 1 to continue or 0 to stop: 1

Enter the packet size: 100

The packet of size 100 is added and in the bucket

## Output:

```
C:\Users\Hp\OneDrive\Desktop\leakybucket.exe
Enter the Bucket size = 5000
Enter the outgoing rate = 200

Enter the packet size = 3000
The Packet of size 3000 is added and in the bucket

Enter 1 to Continue or 0 to Stop: 1

Enter the packet size = 2000
The Packet of size 2000 is added and in the bucket

Enter 1 to Continue or 0 to Stop: 1

Enter the packet size = 1500
The Packet of size 1500 is dropped due to lack of space in the bucket

Enter 1 to Continue or 0 to Stop: 1

Enter the packet size = 100
The Packet of size 100 is added and in the bucket

Enter 1 to Continue or 0 to Stop:
```