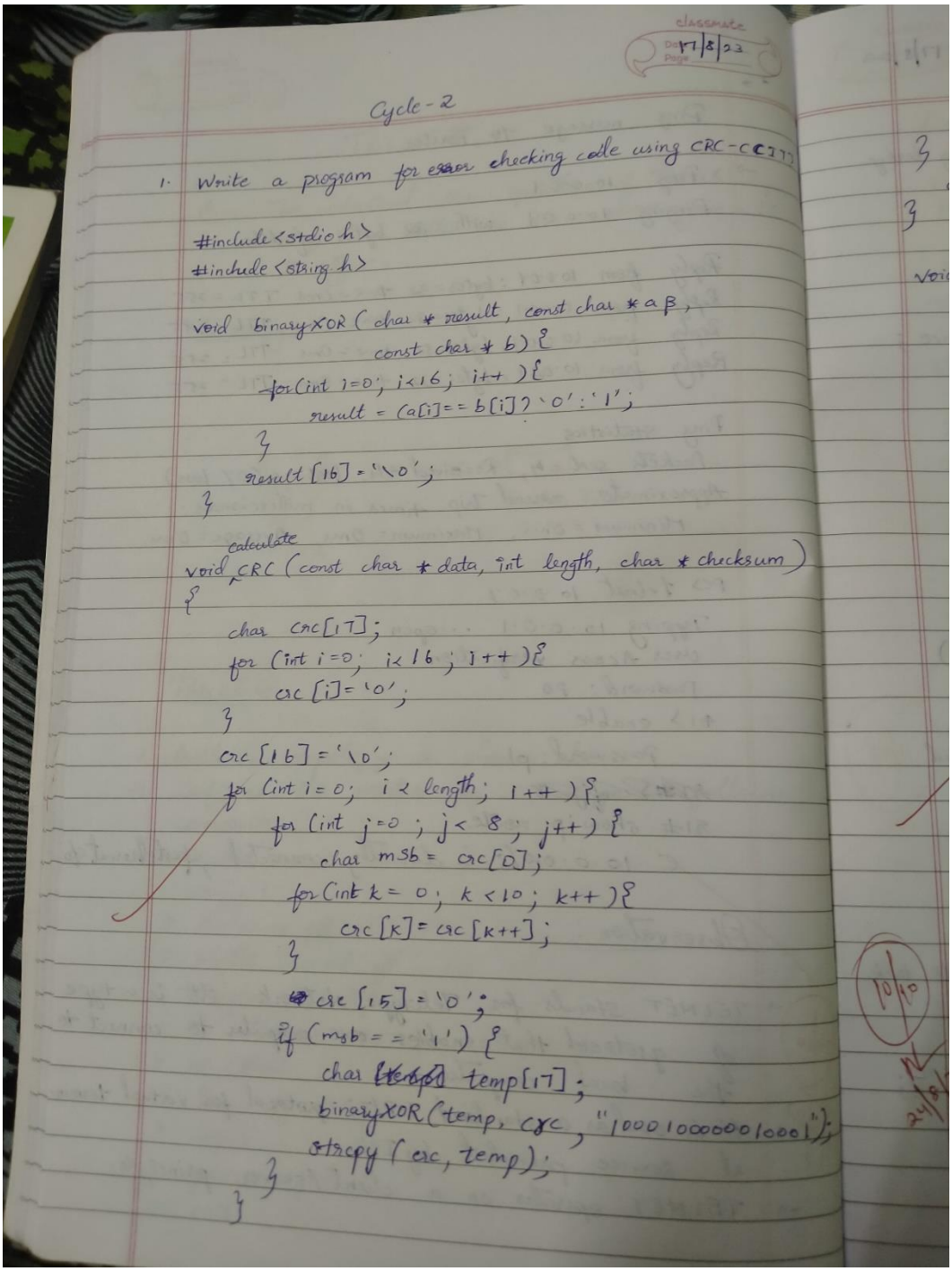
Cycle 2

Experiment 13:

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



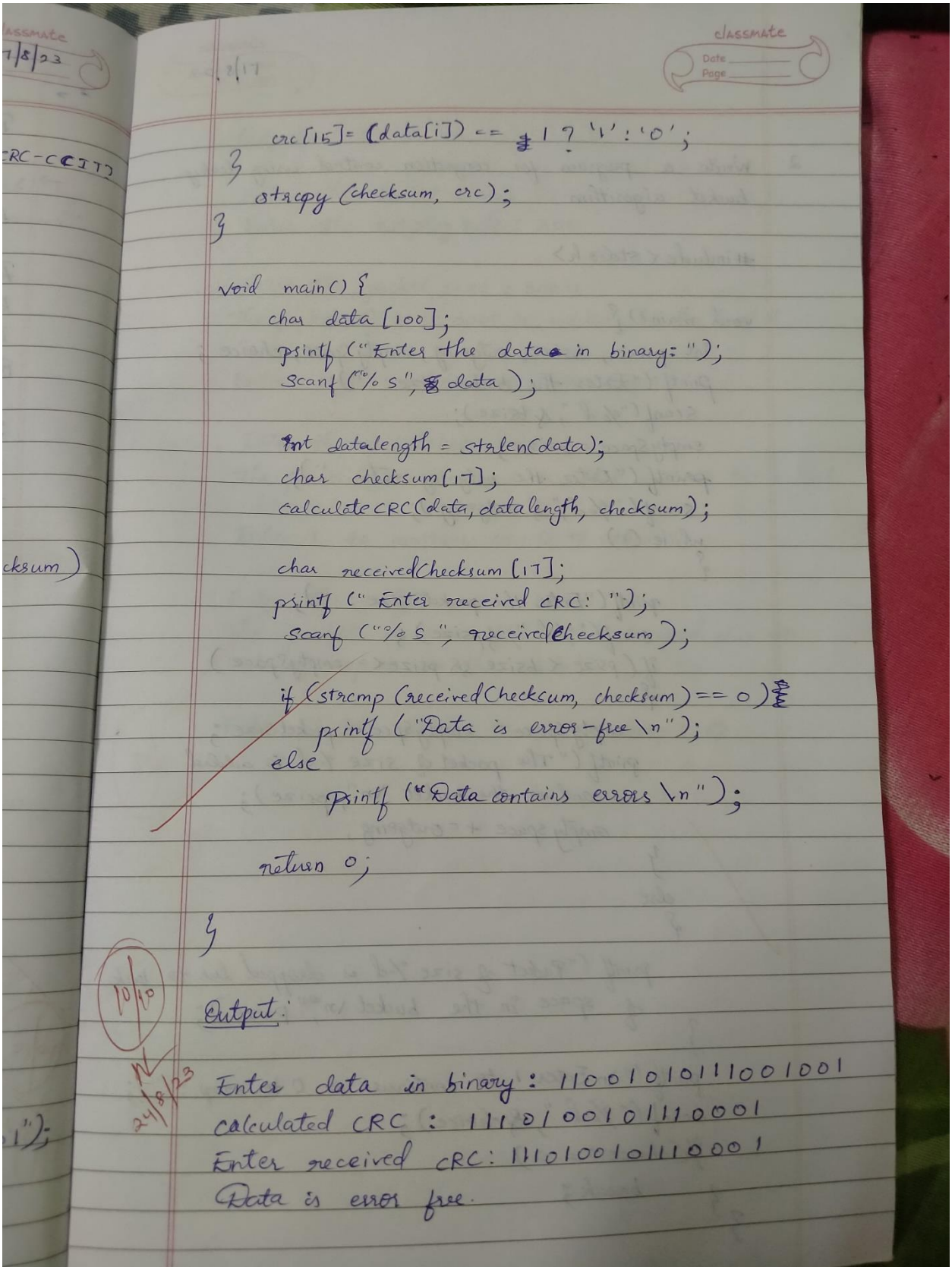
Cycle-2

1. Write a program for error checking 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 = (a[i]==b[i])?'0':'1';
    }
    result[16]='\0';
}

void calculateCRC(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++];
            }
            crc[15]='0';
            if(msb=='1'){
                char temp[17];
                binaryXOR(temp, crc, "1000100000010001");
                strcpy(crc, temp);
            }
        }
    }
```



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

void main() {
    char data[100];
    printf("Enter the 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 : 1100101011001001
Calculated CRC : 1110100101110001
Enter received CRC: 1110100101110001
Data is error free.
```

Screenshots:

crc.c > calculateCRC(const char *, int, char *)

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

// Function to perform bitwise XOR on binary strings
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';
}

// Function to calculate CRC-CCITT checksum
void calculateCRC(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, "10001000000100001"); // CRC_POLY in binary
        strcpy(crc, temp);
      }
    }
    crc[15] = (data[i] == '1') ? '1' : '0';
  }

  strcpy(checksum, crc);
}

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

```
41
42  int main()
43  {
44      char data[100];
45      printf("Enter data in binary: ");
46      scanf("%s", data);
47
48      int dataLength = strlen(data);
49      char checksum[17];
50      calculateCRC(data, dataLength, checksum);
51
52      printf("Calculated CRC: %s\n", checksum);
53
54      // Verify the received data
55      char receivedChecksum[17];
56      printf("Enter received CRC: ");
57      scanf("%s", receivedChecksum);
58
59      if (strcmp(receivedChecksum, checksum) == 0)
60          printf("Data is error-free.\n");
61      else
62          printf("Data contains errors.\n");
63      return 0;
64  }
```

```
PS D:\BMSCE\Academics\Semester IV\Computer networks\Lab\CRC 0> ./crc
Enter data in binary: 11001010111001001
Calculated CRC: 1110100101110001
Enter received CRC: 1110100101110001
Data is error-free.
PS D:\BMSCE\Academics\Semester IV\Computer networks\Lab\CRC 0>
```