COE-342 – Data and Computer Communication
Programming Assignment #3
Due Date: Saturday Jan 18th, 2003

Q.1. Write a program that implements a polynomial divisor to compute the Cyclic Redundancy Check (CRC) code for a given message. The polynomial divisor should be implemented based on the general structure given in the book based on linear feedback shift registers (Figure 7.7 page 207 in textbook). The program should read from a file the pattern P specifying the divisor, and the message M for which the frame check sequence is to be computed. The transmitted message should be generated in another file including the pattern P. The same program should be used to check whether the received message is in error or not. Use a flag in the file to indicate whether the program should compute the FCS of the message or check whether the message is in error or not. For example, if the flag is 0, the program computes the FCS for the transmitted message and appends it to it. Otherwise, it assumes that the message in the file is the received message and reports whether it is received correctly or in error.

Send the **source code** (including *thorough* comments) and the **executable program** by email to: akhayyat@ccse.kfupm.edu.sa and ashraf@ccse.kfupm.edu.sa by the due date. Also, include a **Readme.doc** file on how to run the program and the expected input. Compress all the files to be submitted in a file named **coe342_prog3_<yourID>.zip** that you send by email. The subject in the email should be **"coe342_prog3"**.

Q.2. Use the examples given in the book and notes to verify the correctness of your implementation. Verify **both** the generation of the FCS and that the transmitted message will have a remainder of 0.

Q.3. Using the generator function $P(x) = X^4 + X^2 + 1$, and the message M=1 0 1 0 0 0 1 1 0 1 of length 10 bits, **complete** the following table and **state your observations** on the relation between the percentage of undetected errors and the error type:

Error Type	No of ALL possible Errors	No of detected errors	No of undetected errors	Percentage of undetected errors
ALL single bit errors	14			
ALL double bit errors	91			
ALL three bit errors	364			
ALL four bit errors	1001			

Assume errors can occur in both the message M part of the FCS. Note that for example, the number of all possible single bit errors is equal to 14 since the overall frame length

(M and FCS) is equal to 14 while the number of all 2-bit errors is 14X13/2! = 91. The number of all possible three bit errors is equal to 14X13X12/3! = 364, and so on.

Hint: To complete the table you need to use the routine you developed in Q1 for testing the received frame (i.e after introducing the error) whether it divides completely the generator polynomial or not. The testing should be done for all possible errors of the type of interest (e.g. all single errors, or all 2 bit errors, etc.).
To generate errors in the received frame you will have to generate the error string E, the received frame T containing the errors is then given by: $T_{Rx} = E \oplus T_{Tx}$, where T_{Tx} is the transmitted frame (M + FCS with no errors).
To generate all single bit errors, you need to put the routine you developed in Q1 in a for loop which for each iteration puts a single bit of value 1 in the vector E. For all double bit errors, you will need to use two nested for loops such that the first determines the location of the 1st bit in error while the second loop determines the location of the 2nd bit in error. Of course, these two locations have to be different.

Marking Scheme:

Q1 Total = 50 points + 5 bonus points
- 5 points for the complete self-explanatory (thorough) comments {at least one line of comments per each line of code is expected}
- 5 point for complete source code
- 40 points for correctness of execution
- 5 bonus points for ease of use

Q2 Total = 20 points
- 10 points for a well-documented example (text file, word file, or screen captures) of CRC computation
- 5 points for a well-documented example (text file, word file, or screen captures) of error checking of a correct frame (i.e. dividing the M+FCS by the code P and showing that the result is zero)
- 5 points for a well-documented example (text file, word file, or screen captures) of error checking of an erroneous frame (i.e. dividing the M+FCS by the code P and showing that the result is not zero)

Q3 Total = 30 points + 5 bonus points
- 15 points for the code generating all single, double, 3-bit, and 4-bit errors
- 15 for correctness of table entries and the soundness of the stated observations
- 5 bonus points for ease of use and clarity of output

Late Submission:
The due date for submission is **11.59 pm on Saturday, Jan 18th, 2003**. Programs submitted after the due date will have 10% of the assignment maximum grade subtracted for every late day. If the submission is one week late, the grade for the assignment will be 0. (e.g. If you get a grade of 80% in the assignment and it was submitted 2 days later than the due date, you'll only receive a grade of 60%).