Instructor:
Dr. James W. Hong

Computer Science CS357b
Assignment 1

Due: Noon
Friday
January 27, 1995

You are to write two programs 1) a synchronous transmitter, and 2) a synchronous receiver. Both will assume the use of ASCII codes with odd bit parity.

1. The Synchronous Transmitter

You are to create files of bits ('0' and '1' characters) from the input files using synchronous transmission. Each block will consist of 2 SYN characters (ASCII 22), a control character indicating the length of data block followed by maximum of 64 data characters (see the Figure below). Assume that there is no trailing control characters. Every block transmitted must contain 64 data characters except possibly the case where the remainder of the file cannot fill the buffer. Each character will consist of 7 information bits and a parity bit. These bits will be transmitted from the least significant bit (LSB) to the most significant bit (MSB).

SYN	SYN	Data Length	Data (up to 64 characters)
1 char	1 char	1 char	

The input files are text files called "synch?.dat" (which can be found in the /gaul/s0/course/cs357/asn directory. The output files should be called "synch?.out" and saved in your directory called "asn1".

2. The Synchronous Receiver

This program should take the files created by the Transmitter as input, decode a stream of bits and print out the original transmitted message on the user's console (*e.g.*, stdout).

Both programs should have three distinct layers: physical, data link and application. The physical layer should contain routines to handle jobs such as converting a character into a binary bit pattern, calculating and stuffing a parity bit and writing bits into an output file (*i.e.*, transmission of bits). It should also include routines for reading bits from a file (*i.e.*, reception of bits) and converting bits into characters. The data link layer should contain routines for framing (putting two SYN character, a LENGTH character and data into a frame) and deframing (separating those control characters from data characters). The application layer should contain routines to handle tasks such as reading data to be transmitted from input data file in the Transmitter and writing received data into display (or output data file) in the Receiver, as well as any other routines that you feel necessary.

Your programs should be layered, modularized and well commented if a good mark is expected. The following is a tentative marking scheme and what is expected to be handed in for this assignment:

```
1. External Documentation including [5-10 pages]
   - title page
   - a table of contents
   - [20%] system documentation
     - a high-level data flow diagram for each program
     - a list of routines and their brief descriptions
     - implementation details
   - [10%] test documentation
     - how you tested your program
     - testing outputs
   - [10%] user documentation
     - where is the source (username, directory)
     - how to run the program
     - describe parameters (if any)

2. Source Code
   - [40%] correctness
   - [20%] programming style
     - layering
     - modularity
     - readability
     - comments
     - efficiency
```

Make sure to include my and the TAs' usernames in your .rhosts file (*e.g.*, + jwkhong, + beasley, + oneill, + stevens) and state where your program can be found so that we can verify your program (if necessary).