# Computer Science 435
# Project P1: The Common Meeting Time Problem

**Due:** Mon. Sep. 19, at the beginning of class

The Common Meeting Time problem is the problem of finding a list of the times when a group of people are all available to meet. Write a concurrent program to find the solution to the common meeting time problem for three people.

**Input:** The input to your program will be a file containing three lines of data. The name of the file will be specified as the first and only argument on the command line. Each line of the file will consist of an unsigned integer, followed by zero or more integers: the first number will tell you how many integers follow on that line. Each of the integers after the first represents a "time" when that person is available to meet. You can assume that no time is listed more than once in a list, but do not make any more assumptions about the input.

For example, suppose that the first person can meet at times 3, 2, 4, 1 and 5; the second can meet at 2, and the third can meet at 9, 13, 17, 6, 5, 4, 3 and 2. This information would be represented in the file like this:

```
5 3 2 4 1 5
1 2
8 9 13 17 6 5 4 3 2
```

You can assume that none of the lists contain more than 1,000,000 values.

**Program Structure:** Structure your program as follows: The main program should be responsible for reading the file into three arrays. You should define a `struct` to hold each array and its size. The first array should be declared as a local to `main`. The second and third arrays should be declared as externals (globals), so they can be shared among the threads. No other variables in your program should be declared as externals.

The main program should create a thread for each value $T$ in the first array. Pass $T$ to each thread as a parameter. Each thread is responsible for searching for its associated time in the other two arrays. If a matching value is found in both arrays, a thread should (atomically) print out the string "`T is a common meeting time.`".

If no thread succeeds in finding a common meeting time, your program should output the string "`There is no common meeting time.`"

**What to turn in:** When you have finished with the program, send an email message to me that contains, as an attachment, a copy of your source file. Include in your email any special instructions that I will need to follow to compile and execute your program on the departmental Linux server sand.

Then, print a hard copy of the source to turn in. You must submit your source hard copy within one class day of your electronic submission to receive full credit. I will use the date of your electronic submission as the official submission time, unless you fail to turn in the required hard copy promptly.xc