Topics: Review and examples

Reading (ML): Make sure you've done all the assigned reading!

Example 1

Write a program segment that determines whether a given integer **n** is prime. Assume n>1. (Hint: MATLAB function **mod(x,y)** returns the value of the remainder of x divided by y assuming integer values of x, y.)

Example 2

Sketch a program that will list all the prime numbers in the range of [2,**n**] given an integer **n**>1.

Example 3

Write a program segment that calculates the *cumulative sums* of a given vector **v**. The cumulative sums should be stored in a vector of the same length as **v**. E.g., the cumulative sums for the sequence 1,3,5,0 is 1,4,9,9. Do not use MATLAB predefined functions other than **length**.

Example 4

Develop an algorithm for calculating the *mode* of a sequence. The mode is the number in the sequence that occurs with maximum frequency. Assume that the sequence is (a) non-negative, (b) entered one by one and terminated by a negative number, and (c) entered in non-decreasing order. E.g., the mode of the sequence 87,92,92,98,98,98,100 is 98. Assume that only scalar variables are allowed.

Lecture Survey

Do you have any programming experience? _____

If yes, indicate language and time. _____

Is the lecture pace **too fast**, **about right**, **too slow** ?

Is the lecture content **too difficult**, **about the right level**, **too easy** ?

Lecture would be better if Professor Fan would ...

Lecture would be better if Professor Fan wouldn't ...

Programming Rules of Thumb

- *Learn program patterns* of general utility and *use relevant pattern* for the problem at hand.
- *Seek inspiration* by systematically working test data by hand. Be introspective; ask yourself: "what am I doing?"
- *Declare variables* for each piece of information you maintain when working problem by hand. *Write comments* that precisely describe the contents of each variable.
- *Remember* the problem's boundary conditions.
- *Validate* your program by tracing it on simple test data.

Other comments? (e.g., section, staff, homework, etc.)

Which programming concepts, if any, do you still find confusing?