Example 1: $n$-gon $\rightarrow$ circle

As $n$ increases, the regular inscribed and circumscribed $n$-gons converge to the circle. Since the area of the unit circle is $\pi$, we have

$$\lim_{n \to \infty} \text{innerArea}_n = \pi \quad \lim_{n \to \infty} \text{outerArea}_n = \pi.$$ 

Write a program to find $n$ “sufficiently large” to approximate the area of the unit circle.

```matlab
% Convergence of inner and outer areas.
fprintf('
 n  A(n)  B(n)
');
% Initialization
% Compute and print areas until convergence
while ( )
   fprintf('%4d %9.6f %9.6f 
', n, innerA, outerA);
   innerA = (n/2)*sin(2*pi/n);
   outerA = n*sin(pi/n)/cos(pi/n);
end
```

Syntax of the while Loop

```
while condition
   statements to execute if expression evaluates to true
end
```

If the condition evaluates to true, the loop body executes. Then the condition is evaluated again. When the condition evaluates to false, the loop body is skipped and the program continues after the `end` keyword.

Two useful patterns

<table>
<thead>
<tr>
<th>Pattern for doing something $n$ times</th>
<th>Pattern for doing something an indefinite number of times</th>
</tr>
</thead>
</table>
| for $k = 1 : n$
  % do something
  % ...
| % initialization
  % ...
  while *not stopping signal*
  % do something
  % ...
  % update status (variables)
  % ...
  end |
Example 2: Brute-force algorithm to find minimum function value

How do we find the minimum value of a function \( f(x) \) within some domain \([L, R]\) where \( L < R \)?

Example 3

Write a program fragment that determines whether a given integer \( n \) is prime. Assume \( n > 2 \). (Hint: MATLAB function \( \text{mod}(x,y) \) returns the value of the remainder of \( x \) divided by \( y \) assuming integer values of \( x, y \).)

Example 4

Sketch a program that will list all the prime numbers in the range of \([a, b]\) given integers \( a, b > 1 \) and \( a < b \).

Example 5

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.

The savvy programmer...

- 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.
- Decompose the problem into manageable tasks.
- Refine the algorithm iteratively: solve a simpler problem first
- Remember the problem’s boundary conditions.
- Validate your program by tracing it on simple test data.