Previous Lecture:
- Iteration using while

Today's Lecture:
- Nested loops
- Developing algorithms

Announcements:
- Read Insight §3.2 before discussion, in the lab this week
- Project 2 Parts A & B due Thurs 2/14 at 11pm
- We do not use break in this course
- Make use of Piazza, office hrs, and consulting hrs

What is the last line of output?

```matlab
x = 1;
disp(x)
y = x;
while y==x && x<=4 && y<=4
    x = 2*x;
disp(x)
end
```

A: 1  B: 2  C: 4  D: 8

What is the last line of output?

```matlab
x = 1;
disp(x)
y = x;
while y==x && x<=4 && y<=4
    x = 2*x;
disp(x)
end
```

A: 1  B: 2  C: 4  D: 8

Example: Nested Stars

```m
nestedStars.m
```

Knowing how to draw

```m
Pattern for doing something n times

```

```m
n = _____
for k = 1:n
    % code to do
    % that something
end
```
Example: Are they prime?

- Given integers \(a\) and \(b\), write a program that lists all the prime numbers in the range \([a, b]\).
- Assume \(a>1\), \(b>1\) and \(a<b\).

Subproblem: Is it prime?

- Given integers \(a\) and \(b\), write a program that lists all the prime numbers in the range \([a, b]\).
- Assume \(a>1\), \(b>1\) and \(a<b\).
- Write a program fragment to determine whether a given integer \(n\) is prime, \(n>1\).
- Reminder: \(\text{rem}(x,y)\) returns the remainder of \(x\) divided by \(y\).

Example: Times Table

Write a script to print a times table for a specified range.

<table>
<thead>
<tr>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>9</td>
<td>12</td>
<td>15</td>
<td>18</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>16</td>
<td>20</td>
<td>24</td>
</tr>
<tr>
<td>5</td>
<td>15</td>
<td>20</td>
<td>25</td>
<td>30</td>
</tr>
<tr>
<td>6</td>
<td>18</td>
<td>24</td>
<td>30</td>
<td>36</td>
</tr>
<tr>
<td>7</td>
<td>21</td>
<td>28</td>
<td>35</td>
<td>42</td>
</tr>
</tbody>
</table>

Developing the algorithm for the times table

- Look for patterns
  - Each entry is \(\text{row} \times \text{col}\)
  - \(\text{Row}, \text{col}\) increase regularly
  - \(\Rightarrow\) Loop!!!
- What kind of loop?
  - \(\text{for-loop}\)—since the range of the headings will be specified and increment regularly
  - for each row, get the products with all the cols. Then go to next row and get products with all cols, ...
  - \(\Rightarrow\) Nested loops!
- Details: what will be the print format? Don’t forget to start new lines. Also need initial input to specify the range.
disp('Show the times table for specified range')
lo = input('What is the lower bound? ');
hi = input('What is the upper bound? ');

Rational approximation of $\pi$

- $\pi = 3.141592653589793\ldots$
- Can be closely approximated by fractions, e.g., $\pi \approx 22/7$
- Rational number: a quotient of two integers
- Approximate $\pi$ as $p/q$ where $p$ and $q$ are positive integers $\leq M$
- Start with a straightforward solution:
  - Get $M$ from user
  - Calculate quotient $p/q$ for all combinations of $p$ and $q$
  - Pick best quotient $\Rightarrow$ smallest error

Lecture slides

% Rational approximation of pi
M = input('Enter M: ');
% Check all possible denominators
for q = 1:M
    % At this q, check all possible numerators
    for p = 1:M
        % Check all possible numerators
    end
end

Analyze the program for efficiency

- See Eg3_1 and FasterEg3_1 in the book

for a = 1:n
    disp('alpha')
    for b = 1:m
        disp('beta')
    end
end

The savvy programmer...

- Learns useful programming patterns and use them where appropriate
- Seeks inspiration by working through test data “by hand”
  - Asks, “What am I doing?” at each step
  - Sets up a variable for each piece of information maintained when working the problem by hand
- Decomposes the problem into manageable subtasks
  - Refines the solution iteratively, solving simpler subproblems first
- Remembers to check the problem’s boundary conditions
- Validates the solution (program) by trying it on test data