L5. More on Conditionals

Nested if’s

Multiple Alternatives

Recall the if-else “Template”

```plaintext
if boolean expression
    Commands to execute if the expression if TRUE
else
    Commands to execute if the expression if FALSE
end
```

A Warm-Up Question

Variables a, b, and c have whole number values. True or false: This fragment prints “Yes” if there is a right triangle with side lengths a, b, and c and prints “No” otherwise.

```plaintext
if a^2 + b^2 == c^2
    disp('Yes')
else
    disp('No')
end
```

A. True  B. False

```
 a = 5;
b = 3;
c = 4;
if a^2 + b^2 == c^2
    disp('Yes')
else
    disp('No')
end
```

Prints “no” even though we have:

```
a^2 + b^2 == c^2
(a^2+b^2==c^2) || (a^2+c^2==b^2) || (b^2+c^2==a^2)
```

Developing “If” Solutions

Illustrate the thinking associated with the design of if statements.

The methodology of stepwise refinement.

Two examples...
Problem 1

Write a script that solicits a positive integer y and prints the number of days in year y as determined by the Gregorian calendar.

Leap Year Rule

A non-century year is a leap year if it is divisible by 4.

A century year is a leap year only if it is divisible by 400.

Will Need the Built-In Function rem

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>rem(a,b)</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>56</td>
<td>7</td>
<td>0</td>
</tr>
</tbody>
</table>

The value of \(\text{rem}(a,b)\) is the remainder when the value of a is divided by the value of b. (Assume a, b are whole numbers.)

"Pseudocode" Solution

Input y.

If y is not divisible by 100

Use the non-century year rule.

Otherwise

Use the century year rule.

Refine...

\[
y = \text{input}('\text{Enter the Year:}');
\]

if rem(y,100) ~= 0

%% y is not a multiple of 100

Use the non-century rule

else

%% y is a multiple of 100

Use the century rule

end

Refine the If-Box

%% y is not a multiple of 100

Use the non-century rule

%% y is not a multiple of 100

If y is divisible by 4

Print 366

Otherwise

Print 365
% y is not a multiple of 100
if rem(y,4) == 0
  % y is divisible by 4
  disp('366')
else
  % y is not divisible by 4
  disp('365')
end

% y is divisible by 100
if rem(y,100) ~= 0
  Use the non-century rule
else
  % y is divisible by 400
  Use the century rule
end

y = input('Enter the Year:');
if rem(y,100) ~= 0
  % y is not a multiple of 100
  Use the non-century rule
else
  if rem(y,400) == 0
    disp('366')
  else
    % y is not divisible by 400
    disp('365')
  end
end

y = input('Enter the Year:');
if rem(y,100) == 0
  if rem(y,4) == 0
    disp('366')
  else
    disp('365')
  end
else
  disp('365')
end

The whole thing without comments
Two “Synonyms”

```matlab
if rem(y,4)~=0 || (rem(y,100)==0 && rem(y,400)~=0)
    disp('365')
else
    disp('366')
end
```

```matlab
if rem(y,4)==0 && (rem(y,100)~=0 || rem(y,400)~=0)
    disp('365')
else
    disp('366')
end
```

Key Problem-Solving Strategy

Progress from pseudocode to Matlab through a sequence of refinements.

Comments have an essential role during the transitions. They "stay on" all the way to the finished fragment.

Starting Points Vary In “Friendliness”

A non-century year is a leap year if it is divisible by 4.
A century year is a leap year only if it is divisible by 400.

A year is a leap year if it is divisible by 4 with the exception of century years that are not divisible by 400.

Problem 2

Write a fragment that prints the minimum value of

\[ q(x) = x^2 + bx + c \]

interval.

\[ L <= x <= R \]

One Possibility

\[ q(x) = x^2 + bx + c \]

\[ x_c = -b / 2 \]

Another Possibility

\[ q(x) = x^2 + bx + c \]

\[ x_c = -b / 2 \]
Still Another Possibility

\[ q(x) = x^2 + bx + c \]

\[ x_c = -b/2 \]

We conclude that...

If \( x_c \) is in the interval

The minimum is at \( x_c \)

Otherwise

The minimum is at an endpoint

We Start With Pseudocode...

If \( x_c \) is in the interval

The minimum is at \( x_c \)

Otherwise

The minimum is at an endpoint

Task: Convert to "legal" Matlab.

First refinement...

```matlab
if (L <= xc) && (xc <= R)
  % L <= xc <= R
  % The minimum is at xc.
else
  % xc < L or R < xc
  % The minimum is at an endpoint.
end
```

(1) Boolean expression, (2) commented if-box, (3) commented else box

Refine the If-Box

```matlab
% L <= xc <= R
% The minimum is at xc.

% L <= xc <= R
% The minimum is at xc.
minVal = xc^2 + b*xc + c
```

Refine the Else-Box

```matlab
% xc < L or R < xc
% The minimum is at an endpoint.

xc<L
minVal at L

R<xc
minVal at R
```
% xc < L or R < xc
% The minimum is at an endpoint.

if xc is to the left of L
  The minimum is at L
Otherwise
  The minimum is at R

% xc < L or R < xc
% The minimum is at an endpoint.
if xc < L
  % The minimum is at L
  minVal = L^2 + b*L + c
else
  % The minimum is at R
  minVal = R^2 + b*R + c
end

Overall (w/o Comments)
if (L <= xc) && (xc <= R)
  minVal = xc^2 + b*xc + c.
else
  if xc < L
    minVal = L^2 + b*L + c
  else
    minVal = R^2 + b*R + c
  end
end

Notice there are 3 Alternatives...
if (L <= xc) && (xc <= R)
  minVal = xc^2 + b*xc + c.
else
  if xc < L
    minVal = L^2 + b*L + c
  else
    minVal = R^2 + b*R + c
  end
end

The if-elseif-else Construct
if (L <= xc) && (xc <= R)
  minVal = xc^2 + b*xc + c.
elseif xc < L
  minVal = L^2 + b*L + c
else
  minVal = R^2 + b*R + c
end

Execute exactly one block.

When there are Many Alternatives
if Boolean Expression
  minVal = Boolean Expression
elseif Boolean Expression
  minVal = Boolean Expression
elseif Boolean Expression
  minVal = Boolean Expression
else
  minVal = Boolean Expression
end

Find the first true boolean expression & execute its block. Otherwise execute the else block.
A Common Situation...

```plaintext
if Boolean Expression
    
end
```

When there is nothing to do if the boolean expression is false.