Consider the quadratic function
\[ q(x) = x^2 + bx + c \]
on the interval \([L, R]\):

- Is the function strictly increasing in \([L, R]\)?
- Which is smaller, \(q(L)\) or \(q(R)\)?
- What is the minimum value of \(q(x)\) in \([L, R]\)?

\[
q(x) = x^2 + bx + c \quad \bullet \quad x_c = -b / 2
\]

So what is the requirement?

```matlab
% Determine whether xc is in % [L,R]
xc = -b/2;
if ________________
    disp('Yes')
else
    disp('No')
end
```

The value of a boolean expression is either true or false.

\((L\leq xc) \&\& (xc\leq R)\)

This (compound) boolean expression is made up of two (simple) boolean expressions. Each has a value that is either true or false.

Connect boolean expressions by boolean operators:

and or not
\&\& || ~
Logical operators

&& logical and: Are both conditions true?
E.g., we ask “is \( L \leq x_c \) and \( x_c \leq R \)?”
In our code: \( L \leq x_c \) \&\& \( x_c \leq R \)

|| logical or: Is at least one condition true?
E.g., we can ask if \( x_c \) is outside \([L, R]\), i.e., “is \( x_c < L \) or \( R < x_c \)?”
In code: \( x_c < L \) || \( R < x_c \)

~ logical not: Negation
E.g., we can ask if \( x_c \) is not outside \([L, R]\).
In code: \( \sim (x_c < L \) || \( R < x_c) \)

“Truth table”

| X   | Y   | X \&\& Y | X || Y | \sim Y |
|-----|-----|----------|-------|-------|
| F   | F   | F        |       |       |
| F   | T   | T        |       |       |
| T   | F   | F        |       |       |
| T   | T   | T        |       |       |

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.

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

Always use logical operators to connect simple boolean expressions

Why is it wrong to use the expression
\( L \leq x_c \leq R \)
for checking if \( x_c \) is in \([L, R]\)?

Example: Suppose \( L \) is 5, \( R \) is 8, and \( x_c \) is 10. We know that 10 is not in \([5,8]\), but the expression \( L \leq x_c \leq R \) gives...

Consider the quadratic function
\( q(x) = x^2 + bx + c \)
on the interval \([L, R]\):

- Is the function strictly increasing in \([L, R]\)?
- Which is smaller, \( q(L) \) or \( q(R) \)?
- What is the minimum value of \( q(x) \) in \([L, R]\)?
Start with pseudocode

If \( xc \) is between \( L \) and \( R \)

Min is at \( xc \)

Otherwise

Min is at one of the endpoints

We have decomposed the problem into three pieces! Can choose to work with any piece next: the if-else construct/condition, min at \( xc \), or min at an endpoint.

Set up structure first: if-else, condition

```c
if ( L<=xc && xc<=R )
    Then min is at xc
else
    Min is at one of the endpoints
end
```

Now refine our solution-in-progress. I’ll choose to work on the if-branch next

Refinement: filled in detail for task “min at \( xc \)”

```c
if ( L<=xc && xc<=R )
    % min is at xc
    qMin= xc^2 + b*xc + c;
else
    % min is at one of the endpoints
    if ( xc < L )
        qMin= L^2 + b*L + c;
    else
        qMin= R^2 + b*R + c;
    end
end
```

Continue with refining the solution... else-branch next

Refinement: detail for task “min at an endpoint”

```c
if ( L<=xc && xc<=R )
    % min is at xc
    qMin= xc^2 + b*xc + c;
else
    % min is at one of the endpoints
    if ( xc < L )
        qMin= L^2 + b*L + c;
    else
        qMin= R^2 + b*R + c;
    end
end
```

Continue with the refinement, i.e., replace comments with code

Final solution (given \( b,c,L,R,xc \))

```c
if ( L<=xc && xc<=R )
    % min is at xc
    qMin= xc^2 + b*xc + c;
else
    % min is at one of the endpoints
    if ( xc < L )
        qMin= L^2 + b*L + c;
    else
        qMin= R^2 + b*R + c;
    end
end
```

See quadMin.m
quadMinGraph.m
Notice that there are 3 alternatives \( \rightarrow \) can use elseif!

\[
\begin{align*}
\text{if} & \ L \leq xc \land xc \leq R \\
\text{\% min is at xc} & \\
qMin & = xc^2 + b*xc + c; \\
\text{else} & \\
\text{\% min at one endpt} & \\
\text{if} & \ xc < L \\
qMin & = L^2 + b*L + c; \\
\text{else} & \\
qMin & = R^2 + b*R + c; \\
\end{align*}
\]

Does this program work?

```matlab
score= input('Enter score: '); 
if score>55 
    disp('D')
elseif score>65 
    disp('C')
elseif score>80 
    disp('B')
elseif score>93 
    disp('A')
else 
    disp('Not good…')
end
```

Question

A stick of unit length is split into two pieces. The breakpoint is randomly selected. On average, how long is the shorter piece?

Physical experiment? \( \rightarrow \) analysis

Thought experiment? \( \rightarrow \) analysis

Computational experiment! \( \rightarrow \) simulation

\( \star \) Need to repeat many trials!