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 $[L, R]$?

Modified Problem 3

Write a code fragment that prints “yes” if $x_c$ is in the interval and “no” if it is not.

So what is the requirement?

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

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

$$(L <= x_c) \&\& (x_c <= R)$$

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: \( L = x_c \) \&\& \( x_c = R \)

|| 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: \( \neg (x_c = L) \) || \( R = x_c \)

“Truth table”

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

“Truth table”

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

Logical operators “short-circuit”

A && expression short-circuits to false if the left operand evaluates to false.
A || expression short-circuits to _____________ if ________________

Always use logical operators to connect simple boolean expressions

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

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

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.

```
if \( a^2 + b^2 \) \&\& \( c^2 \)
    disp(’Yes’)
else
    disp(’No’)
end
```

A: true
B: false

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 \( R \)
\[
\text{Min is at } xc
\]
Otherwise
\[
\text{Min is at one of the endpoints}
\]

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

```java
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 \( xc \)”

```java
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
        % xc right of bracket
        % min is at R
end
end
```

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

```java
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
        % xc right of bracket
        % min is at R
end
```

See `quadMin.m` and `quadMinGraph.m`
Notice that there are 3 alternatives → can use elseif!

```matlab
if L<=xc && xc<=R % min is at xc
    qMin= xc^2+b*xc+c;
else % min at one endpt
    if xc < L
        qMin= L^2+b*L+c;
    else
        qMin= R^2+b*R+c;
    end
end

if L<=xc && xc<=R % min is at xc
    qMin= xc^2+b*xc+c;
elseif xc < L
    qMin= L^2+b*L+c;
else
    qMin= R^2+b*R+c;
end
```

Top-Down Design

- State problem
- Define inputs & outputs
- Design algorithm
- Convert algorithm to program

An algorithm is an idea. To use an algorithm you must choose a programming language and implement the algorithm.

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
```

A: yes
B: no