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 \([L, R]\)?

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

\((\text{L} \leq x \leq \text{c}) \land \land (x < \text{c} \leq \text{R})\)

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

\( \land \lor \sim \)

Logical operators

- \&\& logical and: Are both conditions true?
  - E.g., we ask “is \( L \leq x \leq \text{c} \) and \( x < \text{c} \leq R \)?”
  - In our code: \( \text{L} \leq x \leq \text{c} \land \land (x < \text{c} \leq \text{R}) \)

- \| logical or: Is at least one condition true?
  - E.g., we can ask if \( x \leq \text{c} \) is outside \([L, R]\), i.e., “is \( x < L \) or \( R < x \)?”
  - In code: \( x < \text{L} \lor \lor R < x \)

- \~ logical not: Negation
  - E.g., we can ask if \( x \leq \text{c} \) is not outside \([L, R]\).
  - In code: \( \sim (x < \text{L} \lor \lor R < x) \)

Logical operators “short-circuit”

- \&\& condition short-circuits to false if the left operand evaluates to false.
  - A \&\& condition short-circuits to false if the left operand evaluates to false.

- A \| condition short-circuits to false if the first part is false.

```
\begin{tabular}{|c|c|c|c|c|}
\hline
X & Y & X \&\& Y & X \| Y & \sim Y \\
\hline
F & F & F & F & T \\
F & T & F & T & F \\
T & F & F & F & T \\
T & T & T & T & T \\
\hline
\end{tabular}
```
Always use logical operators to connect simple boolean expressions

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

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

\[
\begin{align*}
\text{if } a^2 + b^2 &= c^2 \\
& \text{disp('Yes')} \\
\text{else} \\
& \text{disp('No')} \\
\text{end}
\end{align*}
\]

A: true  
B: false

Conclusion

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

Then min is at \( xc \)

Otherwise

Min value is at one of the endpoints

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

\[
\begin{align*}
\text{if } L <= xc \text{ & } xc <= R \\
& \text{Then min is at } xc \\
\text{else} \\
& \text{Min is at one of the endpoints} \\
\text{end}
\end{align*}
\]

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”

if \( L \leq xc \leq R \)
\[
\begin{array}{l}
\text{\% min is at } xc \\
qMin = xc^2 + b*xc + c;
\end{array}
\]
else
\[
\begin{array}{l}
\text{Min is at one of the endpoints}
\end{array}
\]
end

Continue with refining the solution… else-branch next

Refinement: detail for task “min at an endpoint”

if \( L \leq xc \leq R \)
\[
\begin{array}{l}
\text{\% min is at } xc \\
qMin = xc^2 + b*xc + c;
\end{array}
\]
else
\[
\begin{array}{l}
\text{\% min is at one of the endpoints} \\
\text{if } \% \text{ xc left of bracket} \\
\text{\% min is at } L \\
\text{elseif } \% \text{ xc right of bracket} \\
\text{\% min is at } R \\
\text{end}
\end{array}
\]
end

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

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

if \( L \leq xc \leq R \)
\[
\begin{array}{l}
\text{\% min is at } xc \\
qMin = xc^2 + b*xc + c;
\end{array}
\]
else
\[
\begin{array}{l}
\text{\% min is at one of the endpoints} \\
\text{if } \% \text{ xc L} \\
\text{\% qMin is } L^2 + b*L + c; \\
\text{elseif } \% \text{ xc R} \\
\text{\% qMin is } R^2 + b*R + c;
\end{array}
\]
end

An if-statement can appear within a branch—just like any other kind of statement!

Notice that there are 3 alternatives \( \rightarrow \) can use elseif!

if \( L \leq xc \leq R \)
\[
\begin{array}{l}
\text{\% min is at } xc \\
qMin = xc^2 + b*xc + c;
\end{array}
\]
else
\[
\begin{array}{l}
\text{\% min at one endpt} \\
\text{if } xc < L \\
\text{\% qMin is } L^2 + b*L + c; \\
\text{elseif } xc < R \\
\text{\% qMin is } R^2 + b*R + c;
\end{array}
\]
end

Top-Down Design

State problem
\[\downarrow\]
Define inputs & outputs
\[\downarrow\]
Design algorithm
\[\downarrow\]
Convert algorithm to program
\[\downarrow\]
Decomposition
Stepwise refinement
An algorithm is an idea. To use an algorithm you must choose a programming language and implement the algorithm.