- Previous Lecture (and lab):
  - Variables & assignment
  - Built-in functions
  - Input & output
  - Good programming style (meaningful variable names; use comments)

- Today’s Lecture:
  - Branching (conditional statements)

---

**Announcements:**
- Project 1 (P1) due Thurs, 1/5, at 11pm
- Pay attention to Academic Integrity
- You can see any TA for help, not just your discussion TA
- Matlab consultants at ACCEL Green Rm (Carpenter Hall 2nd fl. computing facility) 5-10pm Sunday to Thursday
- Just added CS1112! Tell your discussion TA to add you in CS1112 CMS (and tell CS1110 to drop your from their CMS)
- Piazza – “Q & A system” for all students in CS1112. Use it for clarification only—do not ask (answer) homework questions and do not give hints on homework. Will be monitored by TAs. Available later today.

---

**Quick review**
- **Variable**
  - A named memory space to store a value
- **Assignment operator:** =
  - Let x be a variable that has a value. To give variable y the same value as x, which statement below should you write?
    - x = y or y = x
- **Script (program)**
  - A sequence of statements saved in an m-file
  - ; (semi-colon)
  - Suppresses printing of the result of assignment statement

---

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 \([L, R]\)?

---

**What are the critical points?**
- End points: \(x = L, x = R\)
- \( \{ x \mid q'(x) = 0 \} \)

\[
q(x) = x^2 + bx + c \\
q'(x) = 2x + b \\
q'(x_c) = 0 \Rightarrow x_c = \frac{-b}{2}
\]
Problem 1

Write a code fragment that prints “yes” if \( q(x) \) increases across the interval and “no” if it does not.

% Quadratic \( q(x) = x^2 + bx + c \)
\[ b = \text{input}('Enter b: '); \]
\[ c = \text{input}('Enter c: '); \]
\[ L = \text{input}('Enter L: '); \]
\[ R = \text{input}('Enter R: '); \]

% Determine whether \( q \) increases
% across \([L,R] \)
\[ xc = -b/2; \]

Problem 2

Write a code fragment that prints “qleft is smaller” if \( q(L) \) is smaller than \( q(R) \).
If \( q(R) \) is smaller print “qright is smaller.”
Algorithm v0
Calculate q(L)
Calculate q(R)
If q(L) < q(R)
    print "qleft is smaller"
Otherwise
    print "qright is smaller"

Algorithm v0.1
Calculate xc
If distance xcL is smaller than distance xcR
    print "qleft is smaller"
Otherwise
    print "qright is smaller"

Do these two fragments do the same thing?
% given x, y
if x>y
    disp('alpha')
else
    disp('beta')
end
% given x, y
if y>x
    disp('beta')
else
    disp('alpha')
end
A: yes  B: no

% Which is smaller, q(L) or q(R)?
qL= L*L + b*L + c;  % q(L)
qR= R*R + b*R + c;  % q(R)
if (qL == qR)
    disp('qleft and qright are equal')
elseif (qL < qR)
    disp('qleft is smaller')
else
    disp('qright is smaller')
end

% Which is smaller, q(L) or q(R)?
qL= L*L + b*L + c;  % q(L)
qR= R*R + b*R + c;  % q(R)
if (qL == qR)
    disp('qleft and qright are equal')
    fprintf('q value is %f\n', qL)
elseif (qL < qR)
    disp('qleft is smaller')
else
    disp('qright is smaller')
end
Consider the quadratic function

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

on the interval \([L, R]\):

What if you only want to know if \(q(L)\) is close to \(q(R)\)?

```matlab
% Is q(L) close to q(R)?

tol= 1e-4; % tolerance
qL= L*L + b*L + c
qR= R*R + b*R + c
if (abs(qL-qR) < tol)
    disp('qleft and qright similar')
end
```

Do these two fragments do the same thing?

```matlab
% given x, y
if  x>y
    disp('alpha')
else
    disp('beta')
end
```

```matlab
% given x, y
if  x>y
    disp('alpha')
end
if  y>=x
    disp('beta')
end
```

A: yes  B: no

Simple if construct

```matlab
if boolean expression
    statements to execute if expression is true
else
    statements to execute if expression is false
end
```

Even simpler if construct

```matlab
if boolean expression
    statements to execute if expression is true
end
```

The if construct

```matlab
if boolean expression1
    statements to execute if expression1 is true
elseif boolean expression2
    statements to execute if expression2 is true
else
    statements to execute if all previous conditions are false
end
```
Things to know about the `if` construct
- At most one branch of statements is executed
- There can be ________ `elseif` clauses
- There can be ________ `else` clause
- The `else` clause ________ in the construct
- The `else` clause ________ (boolean expression)

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 \(xc\) is in the interval and “no” if it is not.

So what is the requirement?

\[
\begin{align*}
\text{\% Determine whether } & xc \text{ is in } \\
\text{\% } [L, R] \text{, } \text{ \%} \\
xc & = -b/2; \\
\text{\%} \\
\text{\%} \text{ end} \\
\end{align*}
\]

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

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

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

\sim \text{ logical not: Negation} \\
E.g., we can ask if \(x_i\) is not outside \([L,R]\). \\
In code: \(\sim (xc \leq L \| R < xc)\)