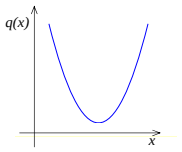
- Previous Lecture:
  - Variables & assignment
  - Input & output
  - Good programming style (meaningful variable names; use comments)
- Today's Lecture:
  - Branching (conditional statements)

- So far, all the statements in our scripts are executed in order
- We do not have a way to specify that some statements should be executed only under some condition
- We need a new language construct…

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 $[L\ ,\ R]$?

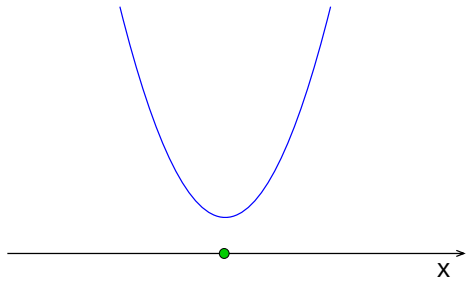

```
% Quadratic q(x) = x^2 + bx + c

b = input('Enter b: ');
c = input('Enter c: ');
L = input('Enter L: ');
R = input('Enter R: ');

% Determine whether q increases
% across [L,R]
```

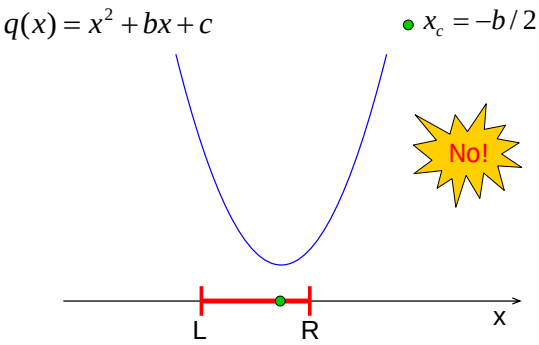

The Situation

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



Problem 1

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



So what is the requirement?

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

if _________________

   disp('Yes')
else
   disp('No')
end
```


Relational operators

<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to
==	Equal to
~=	Not equal to

So what is the requirement?

```
% Determine whether q increases
% across [L,R]
xc = -b/2;
                    fprintf('Yes\n')
if _________
   disp('Yes')
else
   disp('No')
end
```


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 $[L, R]$?

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

Algorithm 1

Calculate q(L)
Calculate q(R)
If q(L) < q(R)
    print "q(L) less than q(R)"
Otherwise
    print "q(R) less than or equal to q(L)"

Algorithm 2

Calculate q(L)
Calculate q(R)
If q(L) is equal to q(R)
    print "q(L) is equal to q(R)"
Otherwise, if q(L) < q(R)
    print "q(L) less than q(R)"
Otherwise
    print "q(R) less than q(L)"

```
% Fragment 1
qL= L*L + b*L + c;  % q(L)
qR= R*R + b*R + c;  % q(R)
if (qL < qR)
   disp('qL less than qR')
else
   disp('qR <= qL')
end
```

```
% Fragment 1.2
qL= L*L + b*L + c;  % q(L)
qR= R*R + b*R + c;  % q(R)
if (qL < qR)
   disp('qL less than qR')
   slope= 2*L + b;
else
   disp('qR <= qL')
   slope= 2*R + b;
end
```

Algorithm 2

Calculate q(L)
Calculate q(R)
If q(L) is equal to q(R)
    print "q(L) is equal to q(R)"
Otherwise, if q(L) < q(R)
    print "q(L) less than q(R)"
Otherwise
    print "q(R) less than q(L)"


```
% Fragment 2

qL= L*L + b*L + c;  % q(L)
qR= R*R + b*R + c;  % q(R)
if (qL == qR)
   disp('qL and qR are the same')
elseif (qL < qR)
   disp('qL less than qR')
else
   disp('qR less than qL')
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 $q(R)$?


```
% Fragment 3
tol= 1e-9;          % tolerance
qL= L*L + b*L + c   % q(L)
qR= R*R + b*R + c   % q(R)
if (abs(qL-qR) < tol)
   disp('qL is close to qR')
end
```

Do these two fragments do the same thing?

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

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

A: yes B: no

Simple `if` construct

if *condition*
  statements to execute if condition is true
else
  statements to execute if condition is false
end

The even simpler `if` construct

if *condition*
  statements to execute if condition is true

end

The general `if` construct

if *condition1*
  statements to execute if condition1 is true
elseif *condition2*
  statements to execute if condition1 is false but condition2 is true
:
else
  statements to execute if all previous conditions are false
end

Rules of the `if` construct

- _____ branch of statements is executed
- _____ **`else`** clause
- _____ **`elseif`** clauses

Consider the quadratic function

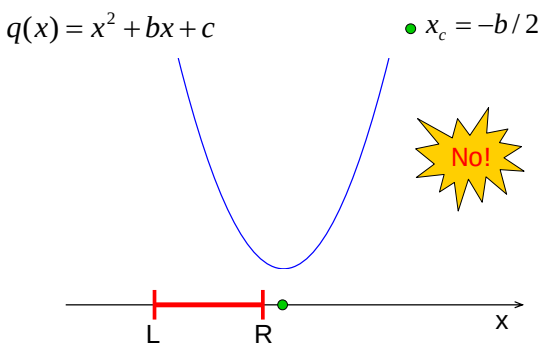
$$q(x) = x^2 + bx + c$$

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?

```
% Determine whether xc is in
% [L,R]
xc = -b/2;

if _________________

   disp('Yes')
else
   disp('No')
end
```


Boolean Expressions

```
(L<=xc) && (xc<=R)
```

Their value is either true or false.

Can be made up of other (simpler) boolean expressions that are connected by boolean operators:

and, or, not

The logical AND operator: &&

F	F	F
F	T	F
T	F	F
T	T	T

The logical OR operator: ||

F	F	F
F	T	T
T	F	T
T	T	T

The logical NOT operator: ~

F	T
T	F