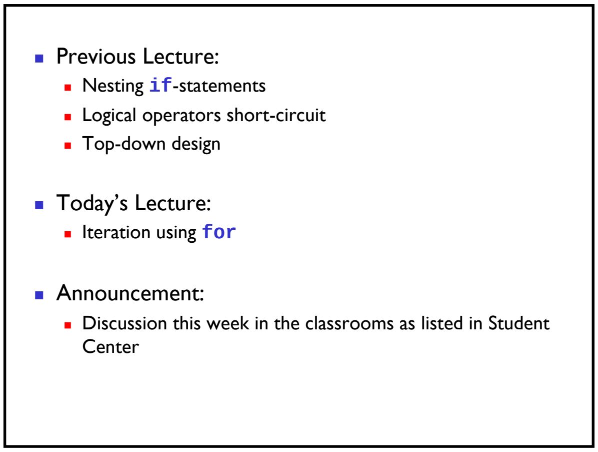
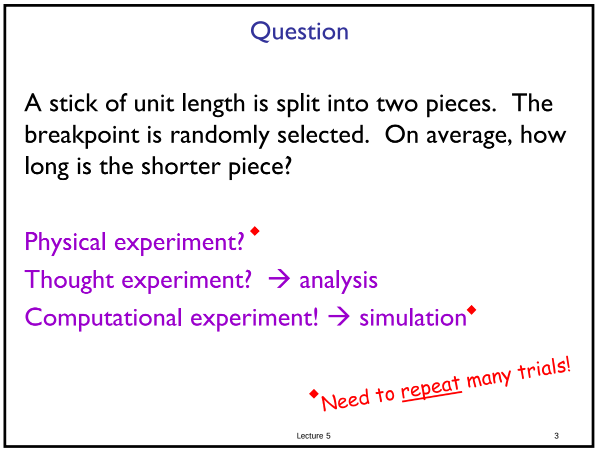


- Previous Lecture:
  - Nesting `if`-statements
  - Logical operators short-circuit
  - Top-down design
- Today's Lecture:
  - Iteration using `for`
- Announcement:
  - Discussion this week in the classrooms as listed in Student Center



Question

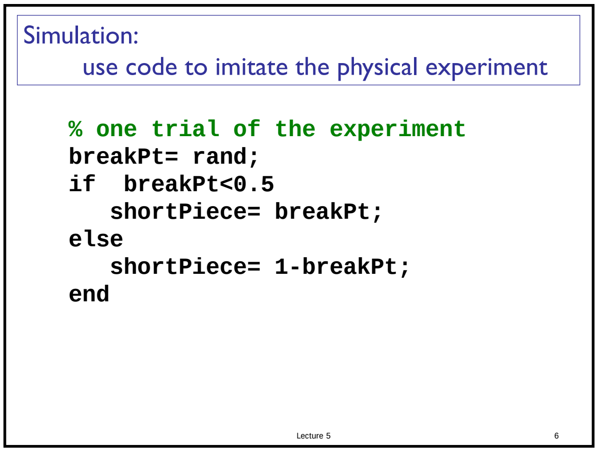
A stick of unit length is split into two pieces. The breakpoint is randomly selected. On average, how long is the shorter piece?

Physical experiment?◆

Thought experiment? → analysis

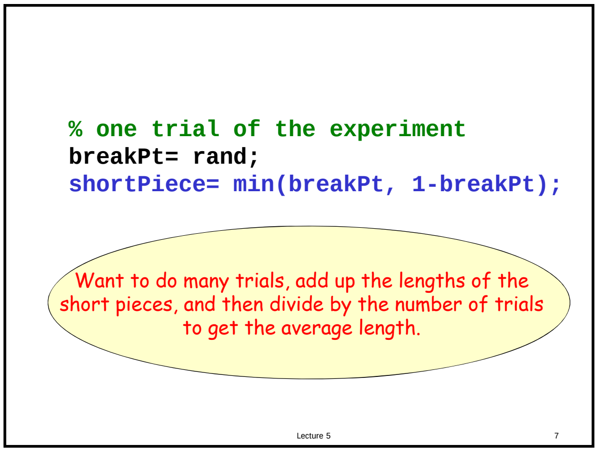
Computational experiment! → simulation◆

◆Need to repeat many trials!



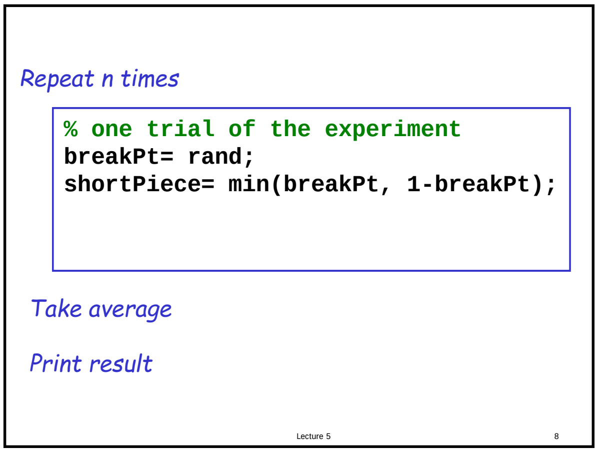
Simulation:
use code to imitate the physical experiment

```
% one trial of the experiment
breakPt= rand;
if  breakPt<0.5
   shortPiece= breakPt;
else
   shortPiece= 1-breakPt;
end
```



```
% one trial of the experiment
breakPt= rand;
shortPiece= min(breakPt, 1-breakPt);
```

Want to do many trials, add up the lengths of the short pieces, and then divide by the number of trials to get the average length.

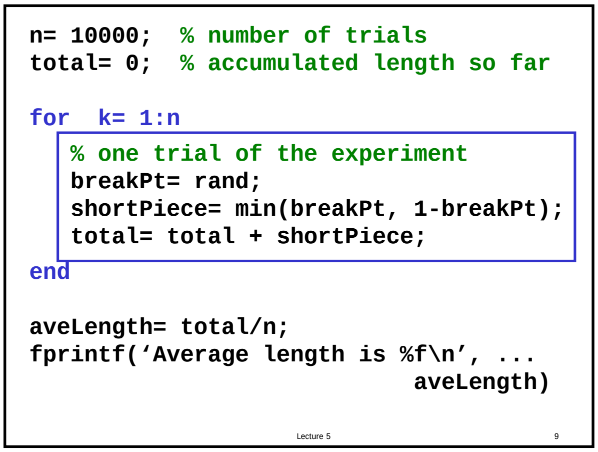


Repeat n times

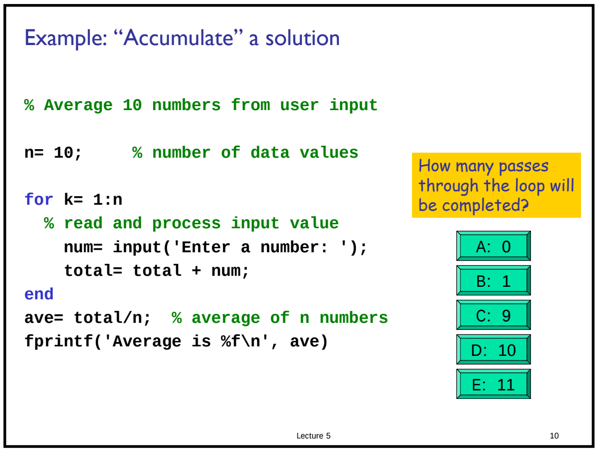
```
% one trial of the experiment
breakPt= rand;
shortPiece= min(breakPt, 1-breakPt);
```

Take average

Print result



```
n= 10000;  % number of trials
total= 0;  % accumulated length so far

for  k= 1:n
   % one trial of the experiment
   breakPt= rand;
   shortPiece= min(breakPt, 1-breakPt);
   total= total + shortPiece;
end

aveLength= total/n;
fprintf('Average length is %f\n', ...
                            aveLength)
```



Example: “Accumulate” a solution

```
% Average 10 numbers from user input

n= 10;     % number of data values

for k= 1:n
  % read and process input value
    num= input('Enter a number: ');
    total= total + num;
end
ave= total/n;  % average of n numbers
fprintf('Average is %f\n', ave)
```

How many passes through the loop will be completed?

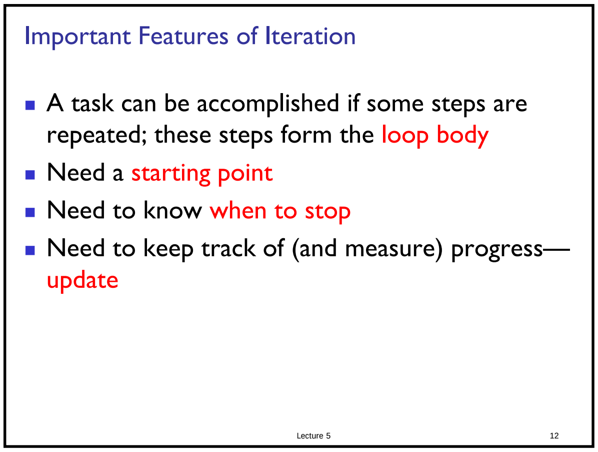
A: 0

B: 1

C: 9

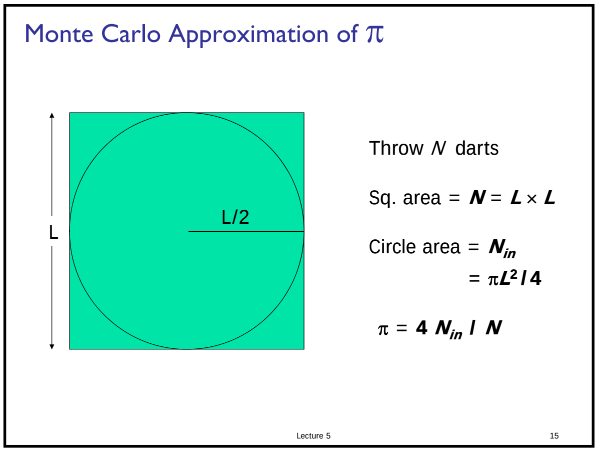
D: 10

E: 11



Important Features of Iteration

- A task can be accomplished if some steps are repeated; these steps form the loop body
- Need a starting point
- Need to know when to stop
- Need to keep track of (and measure) progress—update



Monte Carlo Approximation of π

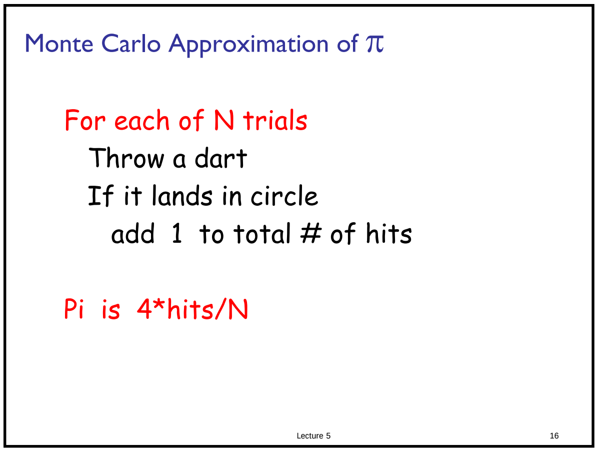
Throw *N* darts

Sq. area = ***N*** = ***L*** × ***L***

Circle area = N_{in}

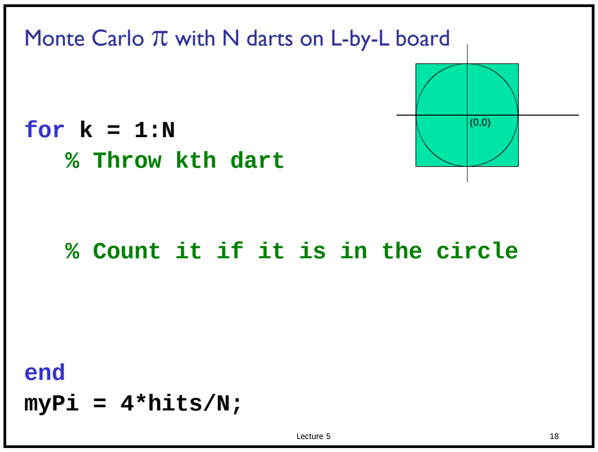
= $\pi L^2 / 4$

$\pi = 4\, N_{in} \,/\, N$



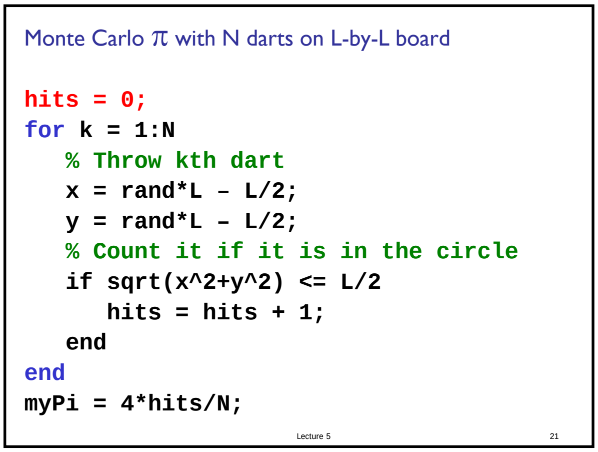
Monte Carlo Approximation of π

```
For each of N trials
  Throw a dart
  If it lands in circle
    add 1 to total # of hits

Pi is 4*hits/N
```



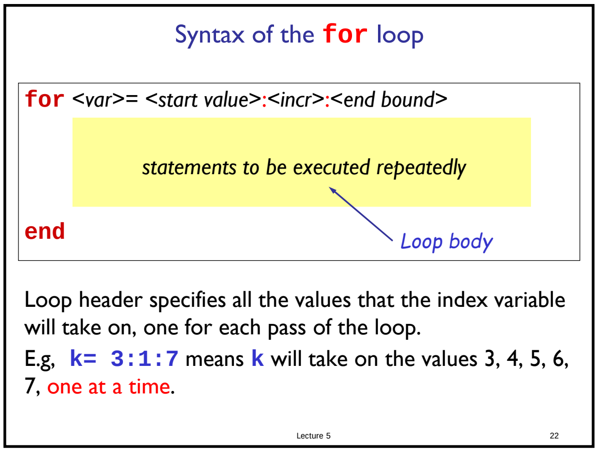
Monte Carlo π with N darts on L-by-L board

```
for k = 1:N
   % Throw kth dart


   % Count it if it is in the circle



end
myPi = 4*hits/N;
```



Monte Carlo π with N darts on L-by-L board

```
hits = 0;
for k = 1:N
   % Throw kth dart
   x = rand*L – L/2;
   y = rand*L – L/2;
   % Count it if it is in the circle
   if sqrt(x^2+y^2) <= L/2
      hits = hits + 1;
   end
end
myPi = 4*hits/N;
```



Syntax of the for loop

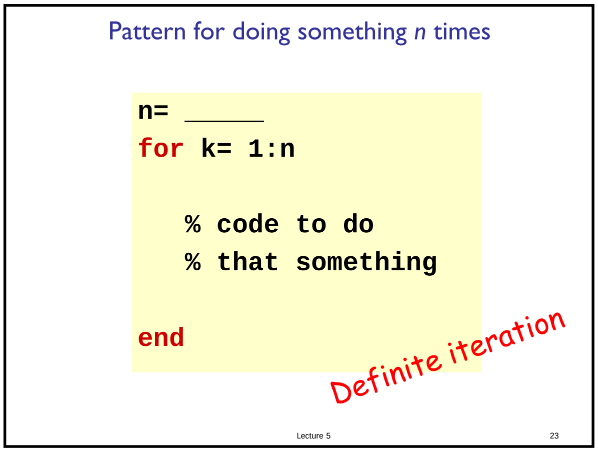
```
for <var>= <start value>:<incr>:<end bound>

        statements to be executed repeatedly

end
```

Loop body

Loop header specifies all the values that the index variable will take on, one for each pass of the loop.

E.g, `k= 3:1:7` means `k` will take on the values 3, 4, 5, 6, 7, one at a time.

Lecture 5 22

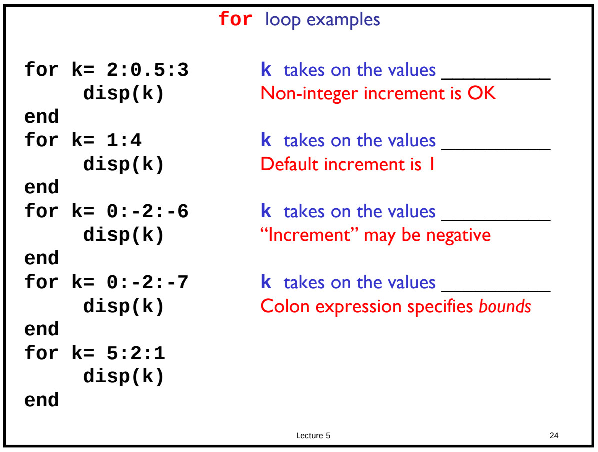


Pattern for doing something *n* times

```
n= ______
for k= 1:n

   % code to do
   % that something

end
```

Definite iteration

Lecture 5 23



for loop examples

```
for k= 2:0.5:3
     disp(k)
end
for k= 1:4
     disp(k)
end
for k= 0:-2:-6
     disp(k)
end
for k= 0:-2:-7
     disp(k)
end
for k= 5:2:1
     disp(k)
end
```

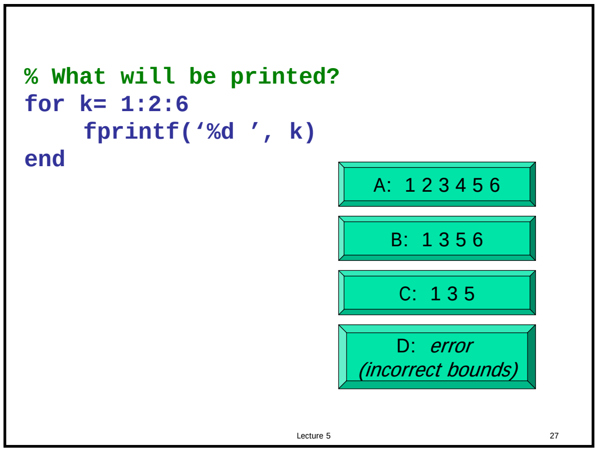
k takes on the values ___________
Non-integer increment is OK

k takes on the values ___________
Default increment is 1

k takes on the values ___________
"Increment" may be negative

k takes on the values ___________
Colon expression specifies *bounds*

Lecture 5 24



```
% What will be printed?
for k= 1:2:6
     fprintf('%d ', k)
end
```

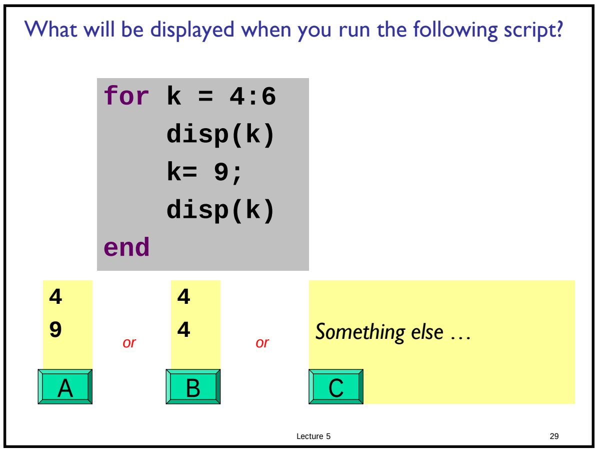
A: 1 2 3 4 5 6

B: 1 3 5 6

C: 1 3 5

D: *error (incorrect bounds)*

Lecture 5 27



What will be displayed when you run the following script?

```
for k = 4:6
    disp(k)
    k= 9;
    disp(k)
end
```

4
9

A

or

4
4

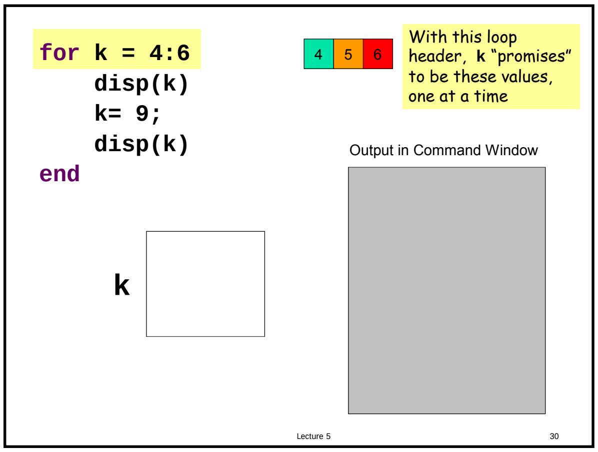
B

or

Something else …

C

Lecture 5 29



```
for k = 4:6
    disp(k)
    k= 9;
    disp(k)
end
```

4	5	6

With this loop header, k "promises" to be these values, one at a time

Output in Command Window

Lecture 5 30