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!

Question

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

A: 0.00001
B: 0.25
C: 0.333333
D: 0.499999
E: none of the above

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

Repeat n times
Take average
Print result

% 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.
\begin{verbatim}
% number of trials
n = 10000;
% accumulated length so far
total = 0;
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
', aveLength)
\end{verbatim}

\begin{verbatim}
% Average 10 numbers from user input
n = 10;     % number of data values
total = 0;  % current sum (initialized to zero)
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
', ave)
\end{verbatim}

\begin{verbatim}
% Monte Carlo Approximation of \pi
for k = 1:N
  % Throw kth dart
  if rand < (L/2)^2/N
    hits = hits + 1;
  end
end
myPi = 4*hits/N;
\end{verbatim}
Syntax of the **for** loop

```
for <var>=<start value>:<incr>:<end bound>
  statements to be executed repeatedly
end
```

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.

**Pattern for doing something \( n \) times**

```
n= _____
for k=1:n
  % code to do
  % that something
end
```

**for** loop examples

```
for k=2:0.5:3  % k takes on the values ______
  disp(k)
end

for k=1:4    % k takes on the values ______
  disp(k)
end

for k=0:-2:-6 % "Increment" may be negative
  disp(k)
end

for k=0:-2:-7 % takes on the values ______
  disp(k)
end
for k=5:2:1  % Colon expression specifies a bound
  disp(k)
end
```

**What will be printed?**

```
for k=10:-1:14
  fprintf('%d ', k)
end
fprintf('!')
```

A: error (Incorrect bounds)
B: 10 (then error)
C: 10 !
D: 14 !
E: !

What will be displayed when you run the following script?

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

A: 4
B: 5
C: 6

Not a condition (boolean expression) that checks whether \( k=6 \).

It is an expression that specifies values:

```
4 5 6
```