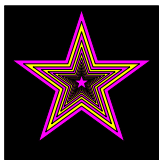
- Previous Lecture:
  - Iteration using `while`
- Today's Lecture:
  - Nested loops
  - Developing algorithms
- Announcements:
  - Project 2 due 9/20 (Mon) at 11pm
  - We do not use `break` in this course
  - Make use of office hrs and consulting hrs
  - Email rbhess@cs.cornell.edu NOW if you have a Prelim 1 conflict

What is the last line of output?

```
x = 1;
disp(x)
y = x;
while y==x && x<=4 && y<=4
   x = 2*x;
   disp(x)
end
```

A: 1 B: 2 C: 4 D: 8

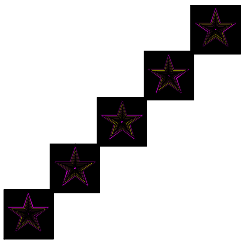
Example: Nested Stars



`nestedStars.m`

Knowing how to draw

How difficult is it to draw



Pattern for doing something *n* times

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

```
x= 0; y= 0; % figure centered at (0,0)

s= 2.1;   % side length of square
DrawRect(x-s/2,y-s/2,s,s,'k')

r= 1;  k= 1;
while  r > 0.1   %r still big
   % draw a star
   if rem(k,2)==1  %odd number
      DrawStar(x,y,r,'m')  %magenta
   else
      DrawStar(x,y,r,'y')  %yellow
   end
   % reduce r
   r= r/1.2;
   k= k + 1;
end
```


Example: Are they prime?

- Given integers a and b, write a program that lists all the prime numbers in the range [a, b].
- Assume a>1, b>1 and a<b.

Example: Are they prime?
Subproblem: Is it prime?

- Given integers a and b, write a program that lists all the prime numbers in the range [a, b].
- Assume a>1, b>1 and a<b.
- Write a program fragment to determine whether a given integer n is prime, n>1.
- Reminder: rem(x,y) returns the remainder of x divided by y.

Developing the algorithm for the times table

	3	4	5	6	7
3	9	12	15	18	21
4	12	16	20	24	28
5	15	20	25	30	35
6	18	24	30	36	42
7	21	28	35	42	49

```
disp('Show the times table for specified range')
lo= input('What is the lower bound? ');
hi= input('What is the upper bound? ');
```

Rational approximation of π

- π = 3.141592653589793…
- Can be closely approximated by fractions, e.g., $\pi \approx 22/7$
- Rational number: a quotient of two integers
- Approximate π as p/q where p and q are positive integers ≤M
- Start with a straight forward solution:
  - Get M from user
  - Calculate quotient p/q for all combinations of p and q
  - Pick best quotient → smallest error


```
% Rational approximation of pi

M = input('Enter M: ');


% Check all possible denominators
```

Analyze the program for efficiency

- See Eg3_1 and FasterEg3_1 in the book

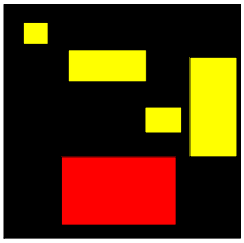
```
for a = 1:n
   disp('alpha')
   for b = 1:m
      disp('beta')
   end
end
```

How many times are "alpha" and "beta" displayed?

- A: n, m
- B: m, n
- C: n, n+m
- D: n, n*m
- E: m*n, m

Find the biggest rectangle

Here's the biggest rectangle you drew!



- Draw 5 rectangles that the user specifies using mouse clicks
- Color the biggest one red

Find the *mode* in a sequence

- A mode is the number in a sequence that appears the most number of times
- Develop an algorithm for calculating the mode of a user-entered sequence that is
  - Non-negative
  - Entered one-by-one in non-decreasing order
  - Terminated by a negative number
- E.g., sequence 87, 92, 92, 98, 98, 98, 100 has a mode…
- Write the algorithm and then the code on your own for practice!

The savvy programmer…

- Learns useful programming patterns and use them where appropriate
- Seeks inspiration by working through test data "by hand"
  - Asks, "What am I doing?" at each step
  - Sets up a variable for each piece of information maintained when working the problem by hand
- Decomposes the problem into manageable subtasks
  - Refines the solution iteratively, solving simpler subproblems first
- Remembers to check the problem's boundary conditions
- Validates the solution (program) by trying it on test data