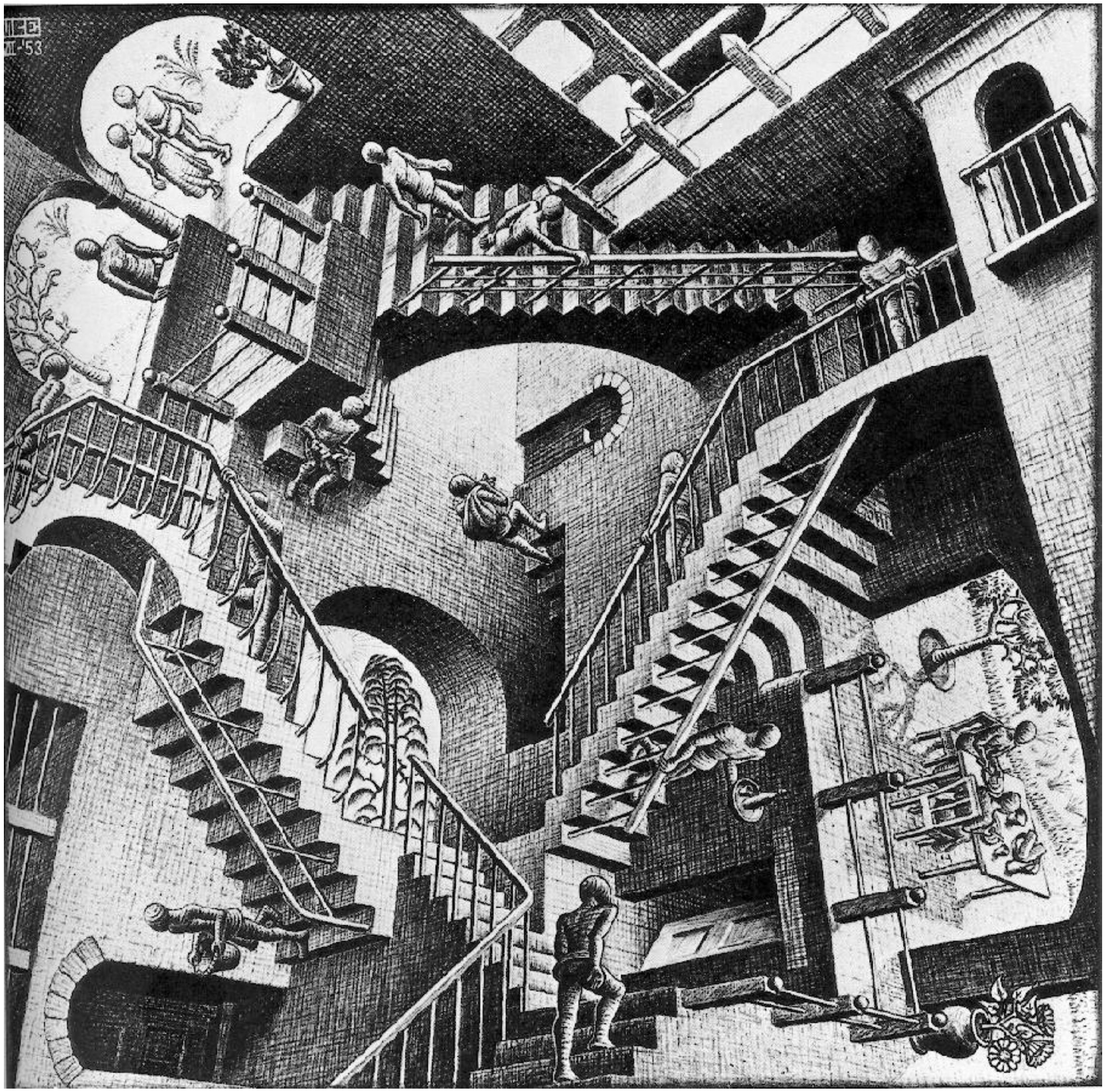


Structures

Lecture 19 (Apr 1)
CS100M – Spring 2008

Announcements

- Section is in the lab this week

- The last project was more challenging than previous ones
  - Problem 2 was more difficult than problem 1
    - Pulling out the bits
    - Working with uint8
  - Some of the problem 1 functions could be *very* brief
    - Could use vectorized code

- Please don't violate Academic Integrity
  - We run a program to detect similar code

- Project 5 should be available late today

Data is Often Related

- A point in the plane has an x coordinate and y coordinate
- If a program manipulates lots of points, there will be lots of x's and y's
  - Anticipate clutter
- Is there a way to "package" the two coordinate values?

Packaging Affects Thinking

- Our Reasoning Level:
  - P and Q are points
  - Compute the midpoint M of the connecting line segment
- Behind the scenes we do this:

$$M_x = (P_x + Q_x)/2$$
$$M_y = (P_y + Q_y)/2$$

- We've seen this before
  - Functions are used to "package" calculations
- This kind of packaging (a type of abstraction) elevates the level of our reasoning
- Critical for problem solving!

Simple Example

```
p1 = struct('x', 3, 'y', 4);

p2 = struct('x', -1, 'y', 7);

D = sqrt((p1.x-p2.x)^2 + (p1.y-p2.y)^2);
```

Distance between two points

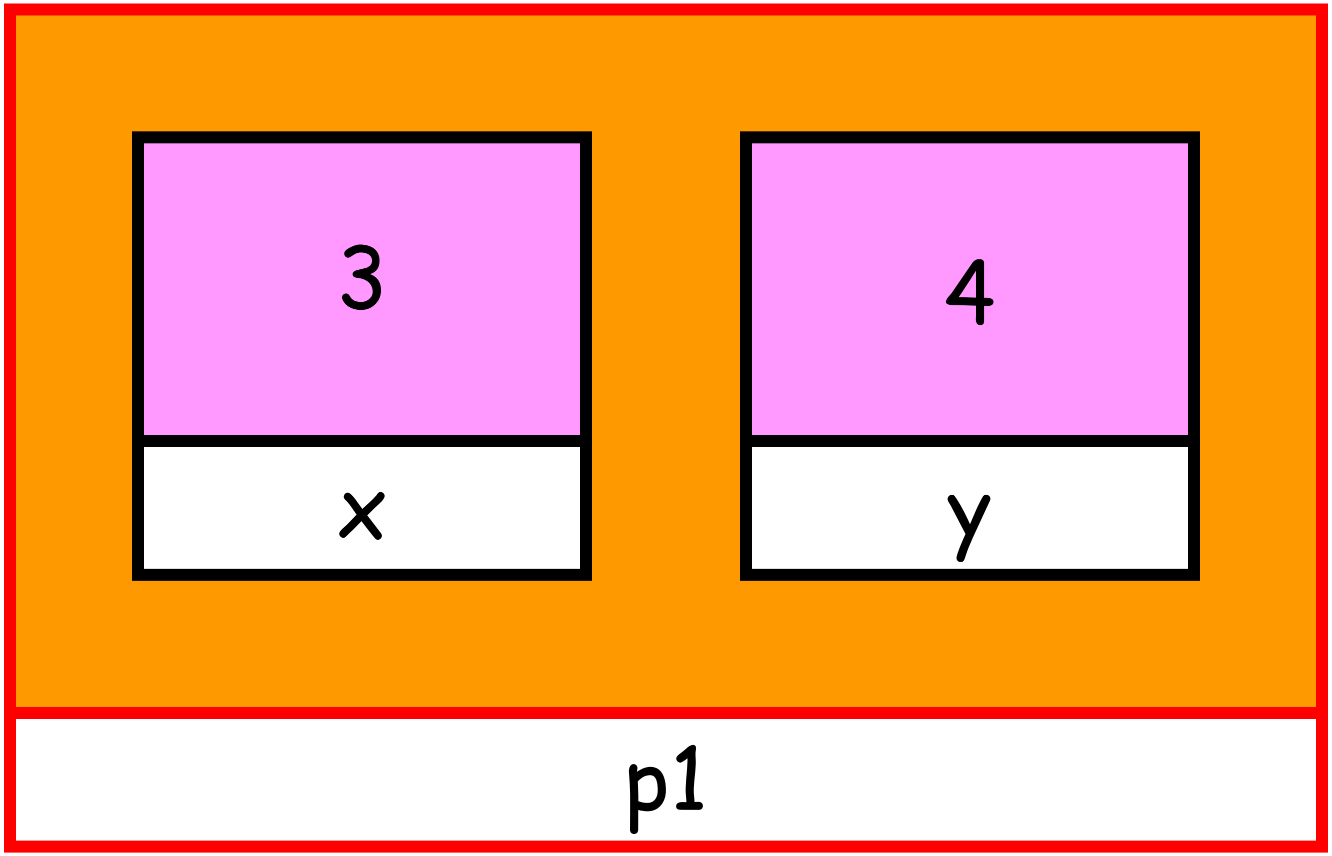
p1.x, p1.y, p2.x, p2.y are participating as variables—because they are

Initialization

```
p1 = struct('x', 3, 'y', 4);
```

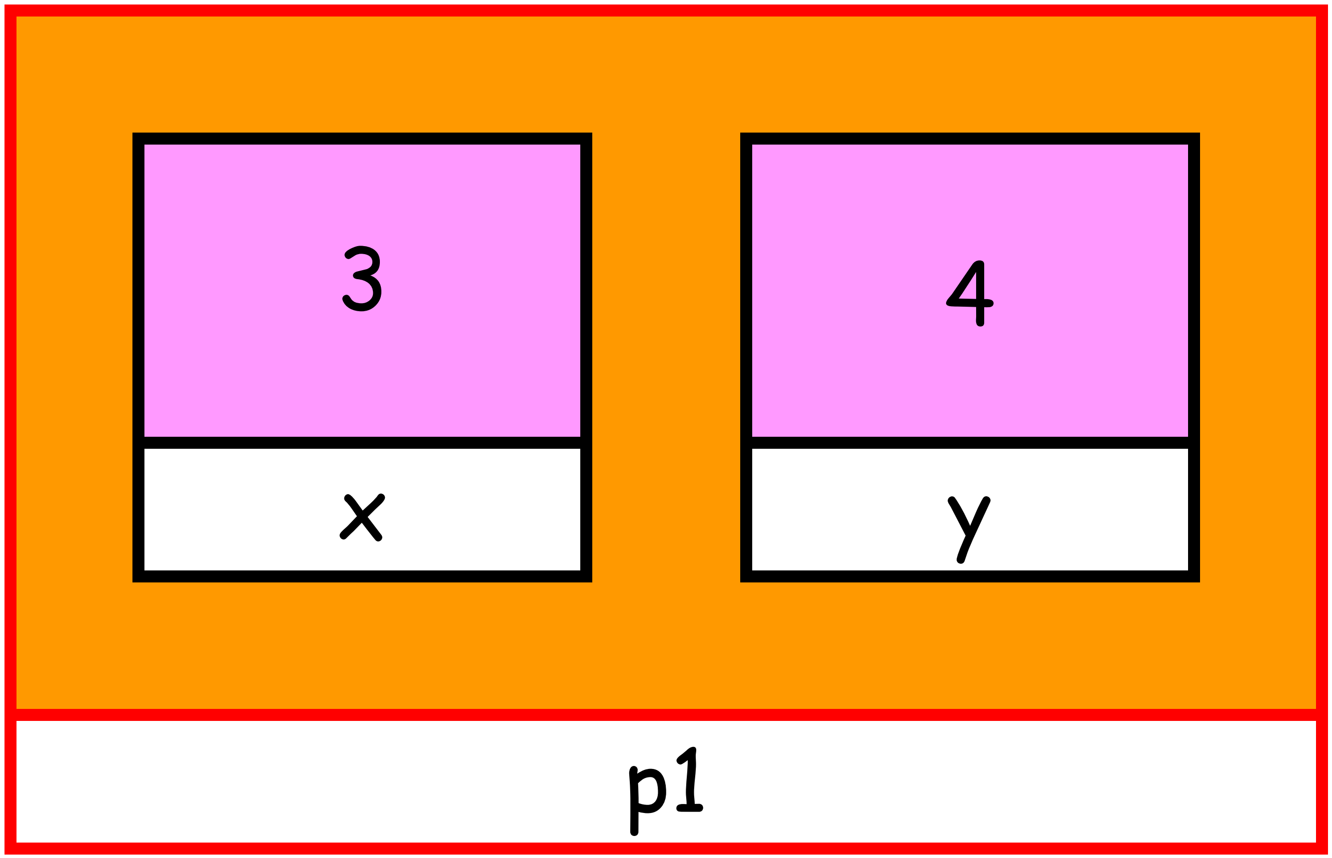
- p1 is a structure
- The structure has two fields
- Their names are x and y
- They are assigned the values 3 and 4

How to Visualize p1



p1 = struct('x', 3, 'y', 4);

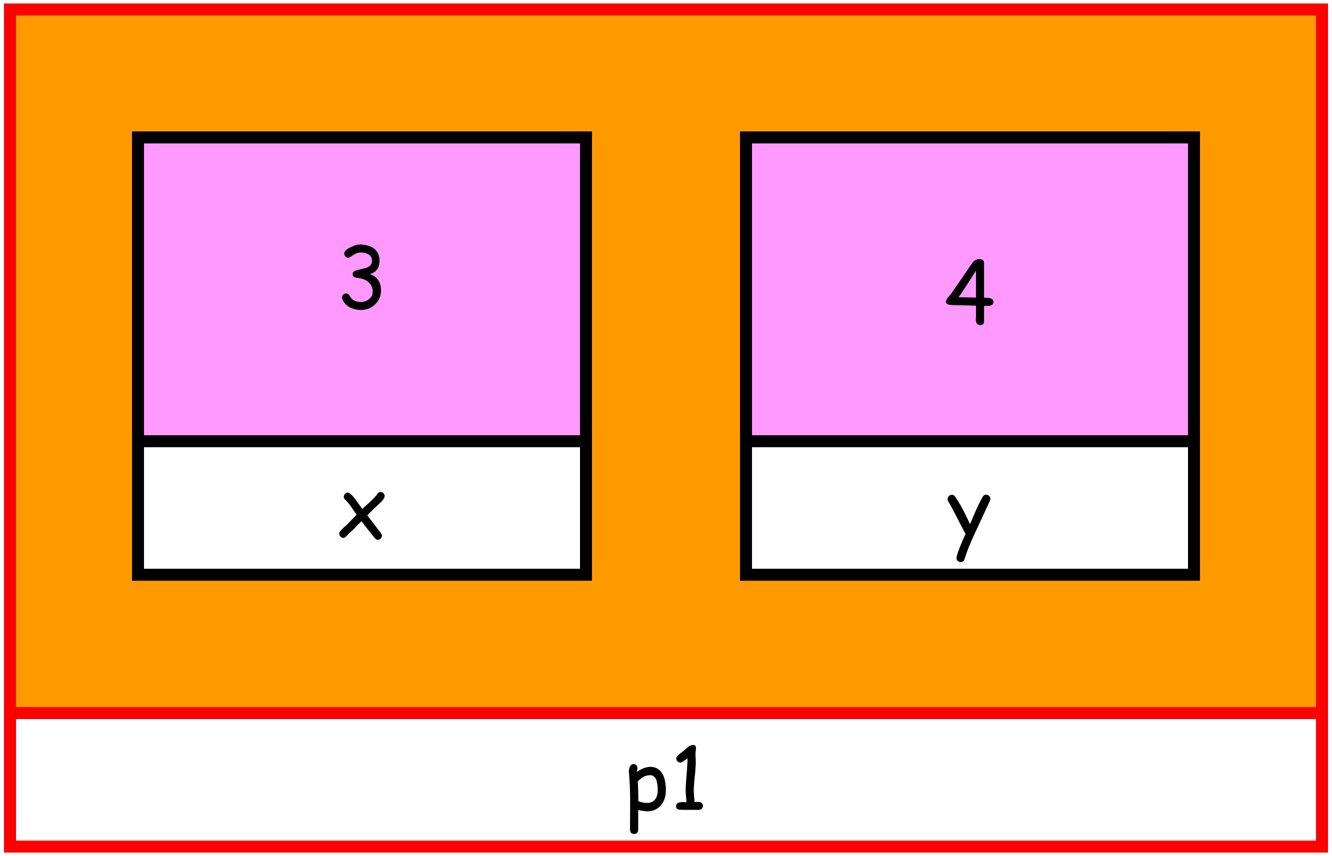
Accessing a Field



A = p1.x + p1.y

Assigns the value 7 to A

Assigning to a Field




```
p1.x = p1.y^2
```

Assigns the value 16 to p1.x

Another Example

```
A = struct('name', 'New York', 'capital', 'Albany', 'Pop', 15.5);
```

- Can have combinations of string fields and numeric fields

Legal/Illegal Maneuvers

```
Q = struct('x', 5, 'y', 6);

R = Q;                        % Legal: R is a copy of Q

S = (Q+R)/2;                  % Illegal: Cannot add structures

P=struct('x', 3, 'y');        % Illegal: Args must be in pairs
P.y = 4;

P = struct('x',3,'y',[]);     % Legal: Empty array as a "place holder"
P.y = 4;
```

Structures in Functions

```
function d = dist(P,Q)
% P and Q are points.
% d is the distance between them

  d = sqrt((P.x-Q.x)^2 + (P.y-Q.y)^2);
```

Sample “Make” Function

```
function P = MakePoint(x,y)
% P is a point.
% P.x and P.y are assigned the values x and y.

P = struct('x',x,'y',y);
```

- Good style
- Highlights the structure’s definition

Functions and Structures

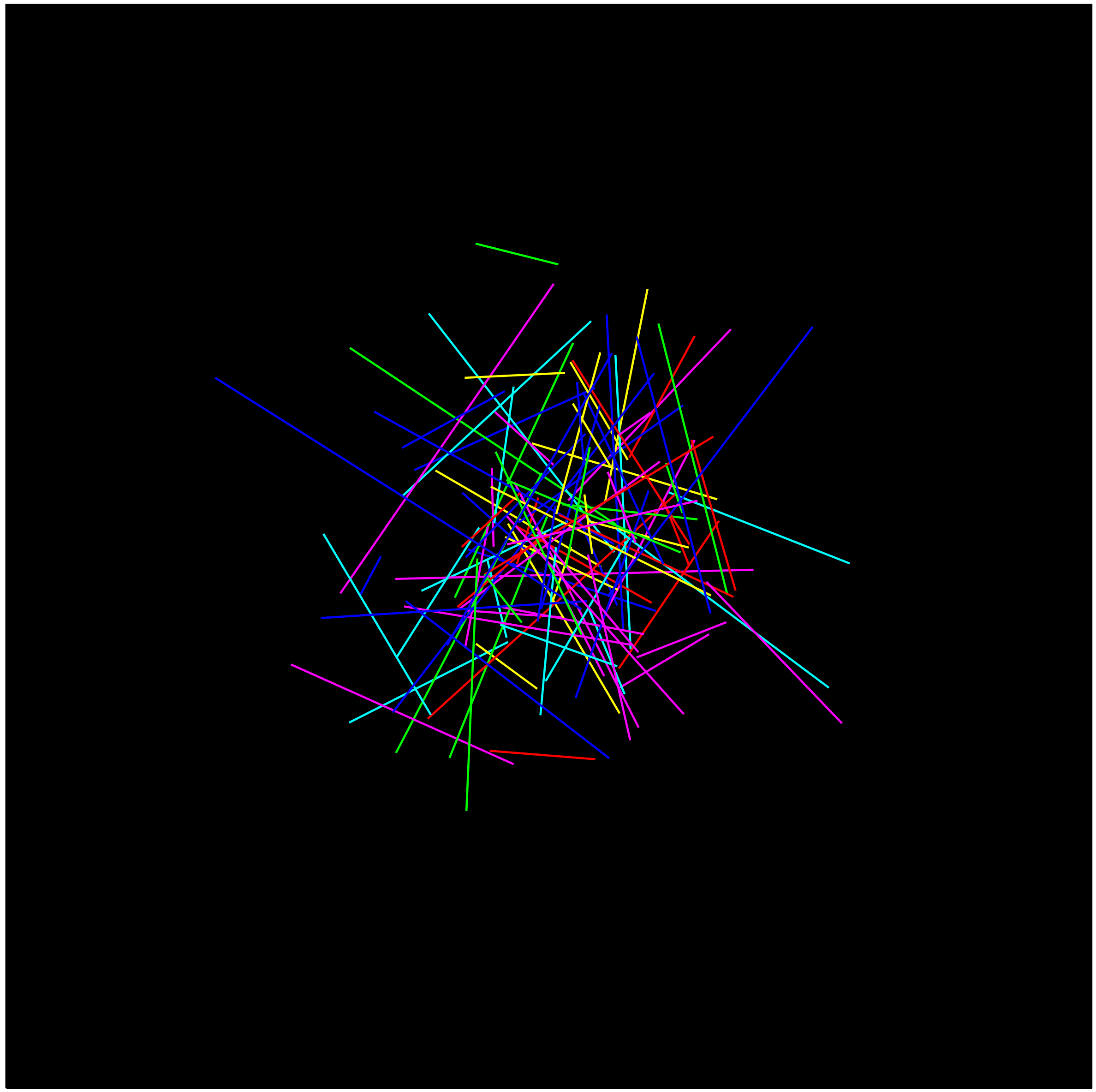
```
function DrawLS (P, Q, c)
% Draws a line segment connecting points
% P and Q; color is specified by c.
% Assumes hold is on.

plot([P.x  Q.x], [P.y  Q.y], c)
```

Script for Pick Up Sticks

```
s = 'rgbmcy';
set(gcf,'color','k')
axis equal off
hold on
for k=1:100
   P = MakePoint(randn,randn);
   Q = MakePoint(randn,randn);
   c = s(ceil(6*rand));
   DrawLS(P,Q,c)
end
```

Generates two random points and chooses one of six colors randomly



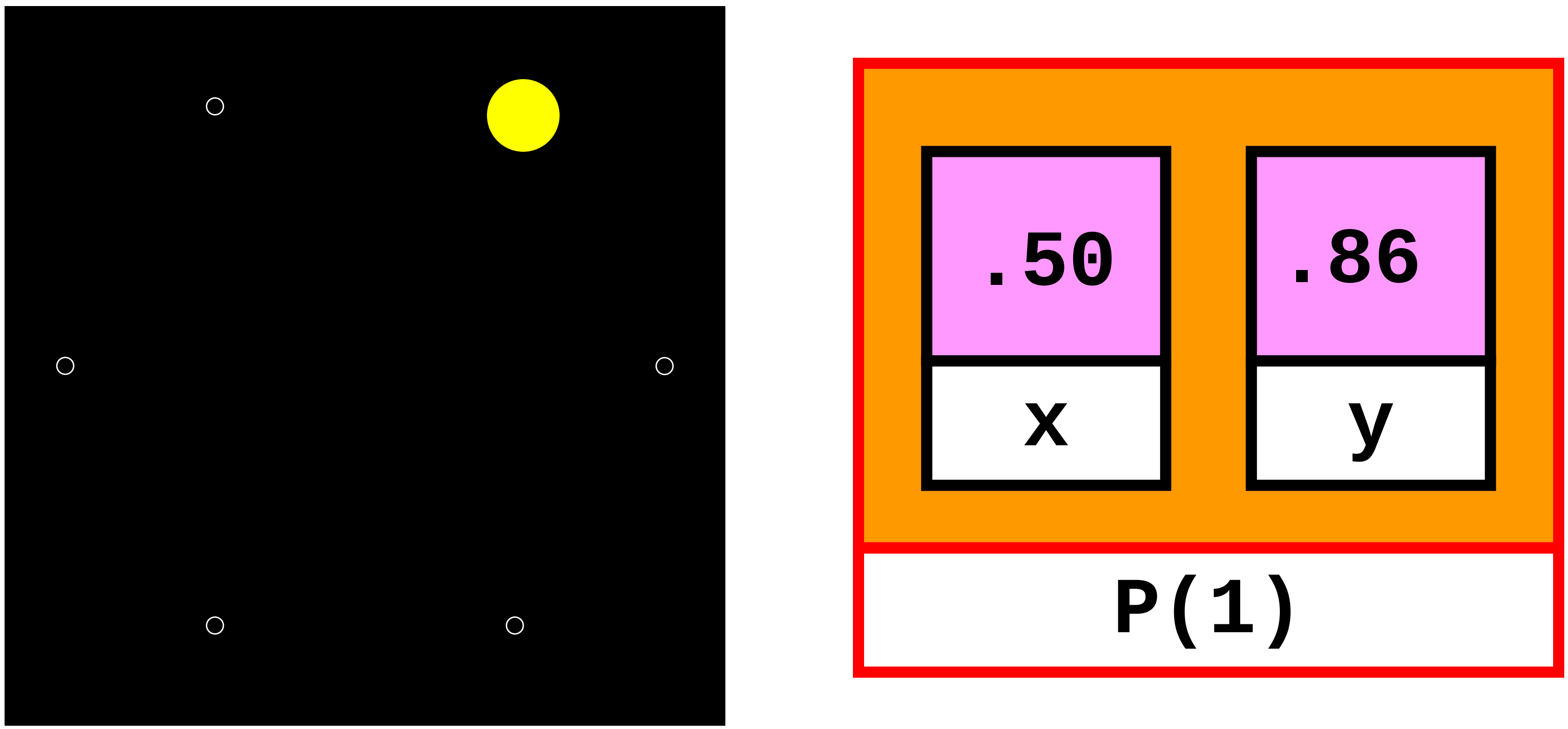
Structure Arrays

- An array whose components are structures

- All the structures must be the same

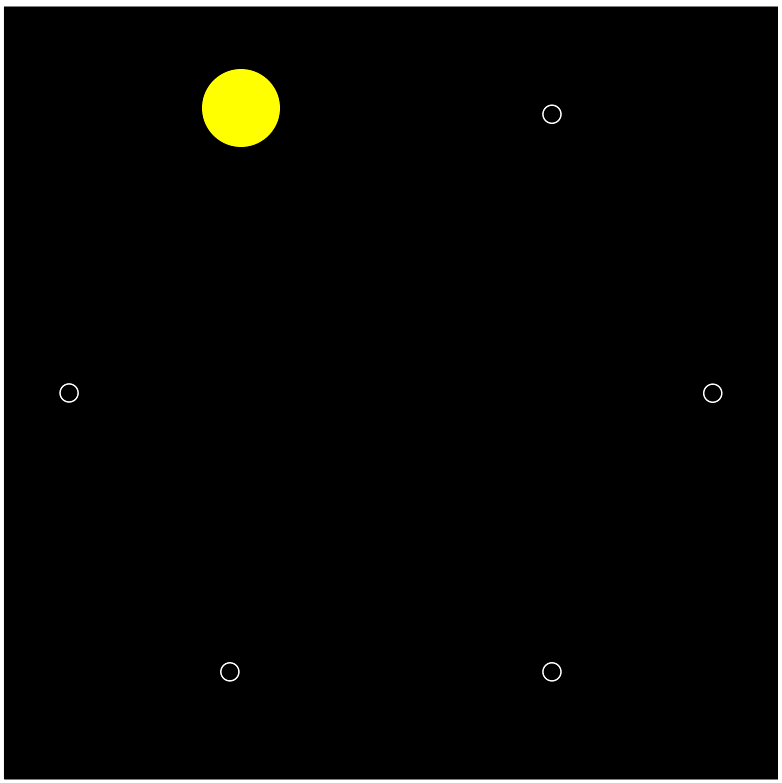
Example: A point array…

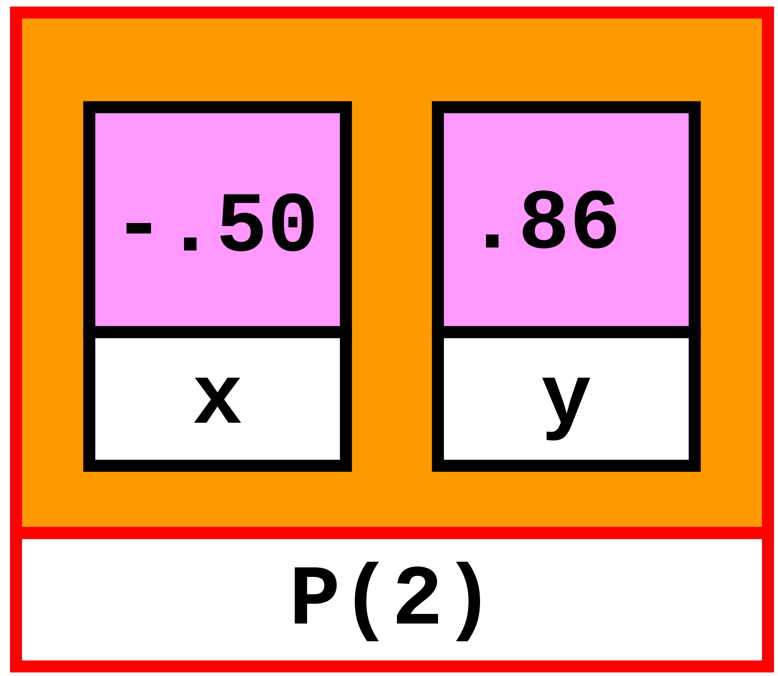
An Array of Points



`P(1) = MakePoint(.50,.86)`

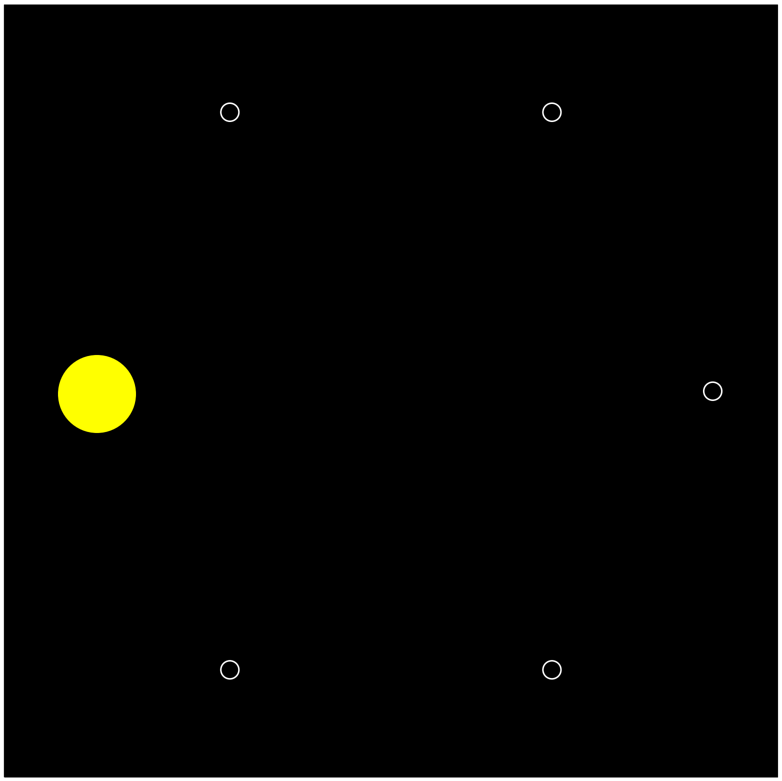
An Array of Points

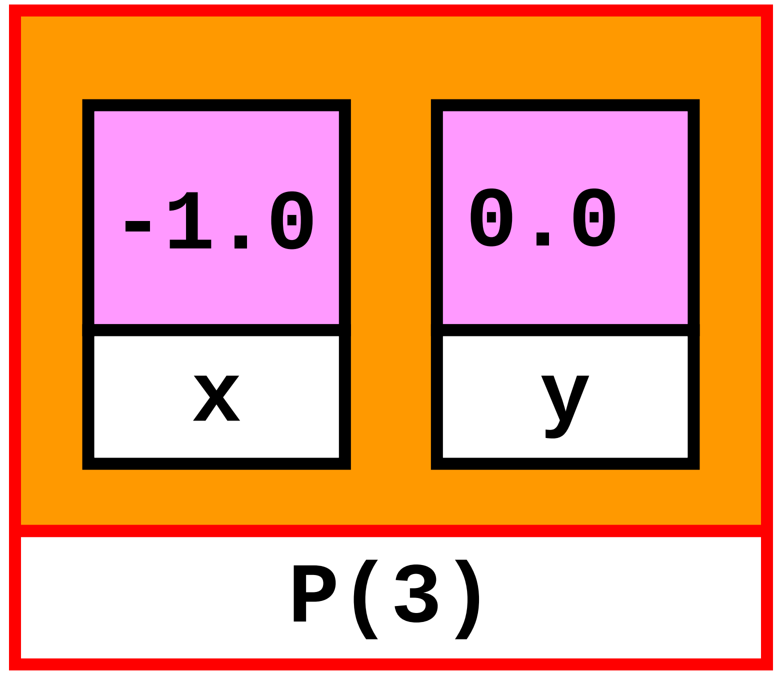





```
P(2) = MakePoint(-.50,.86)
```

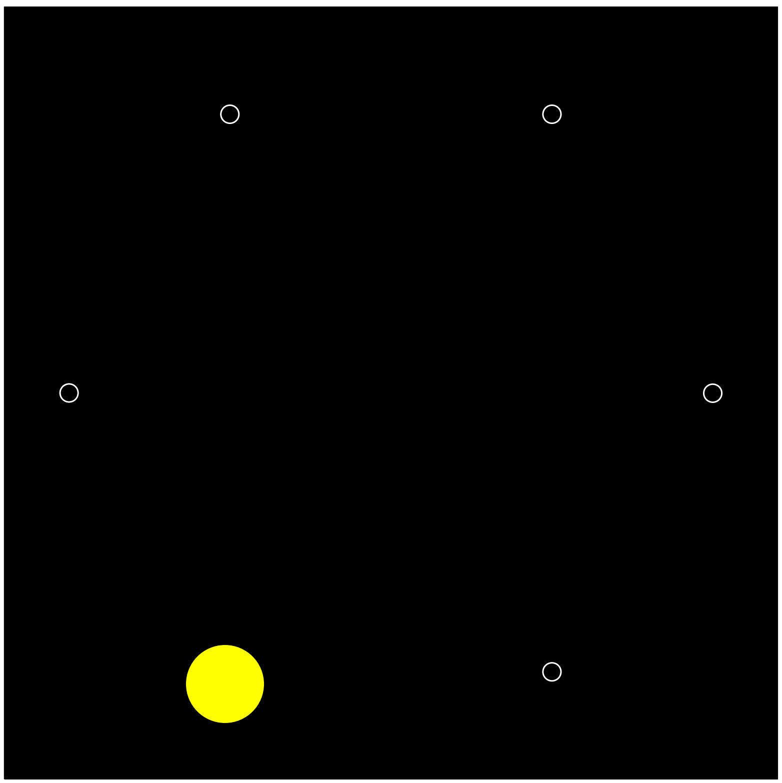
An Array of Points

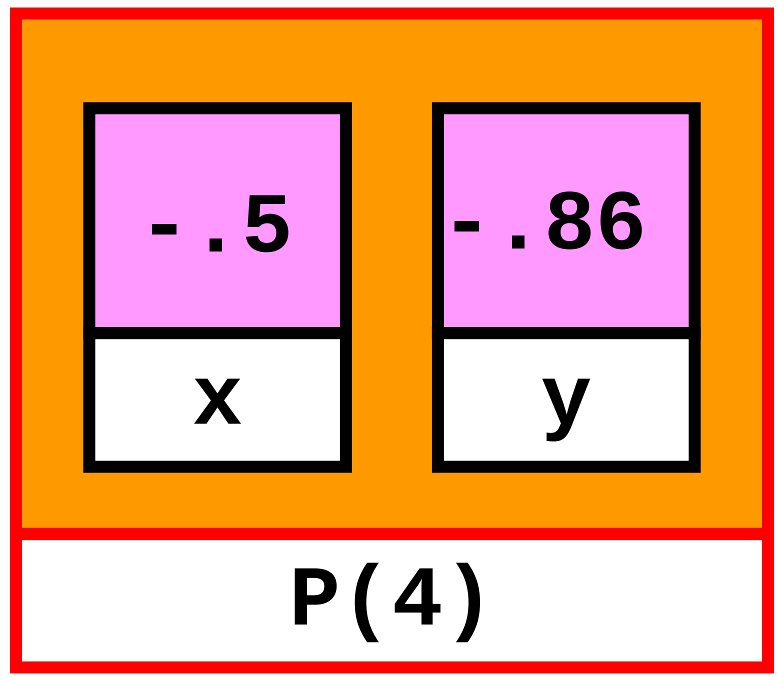





```
P(3) = MakePoint(-1.0,0.0)
```

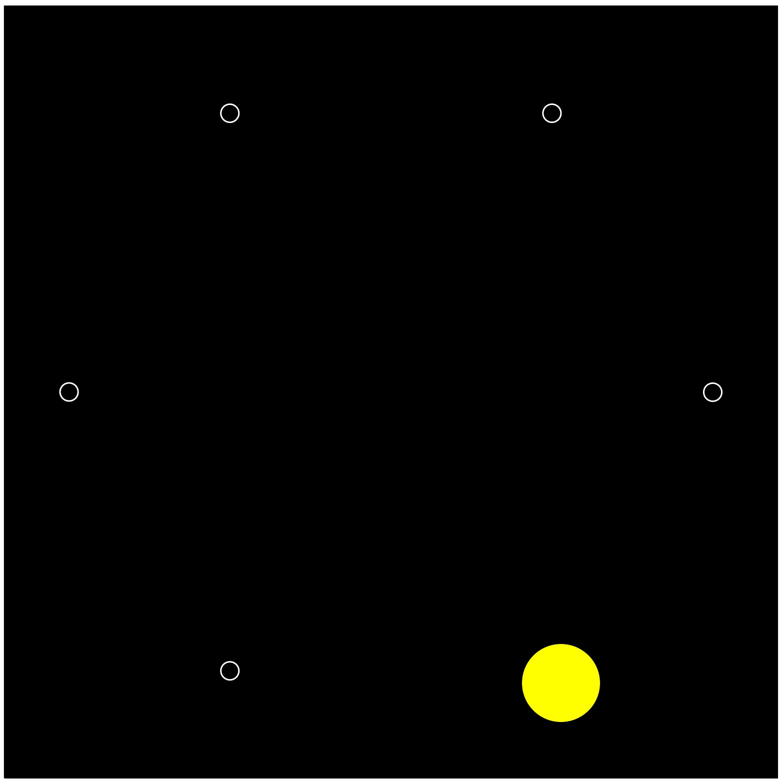
An Array of Points

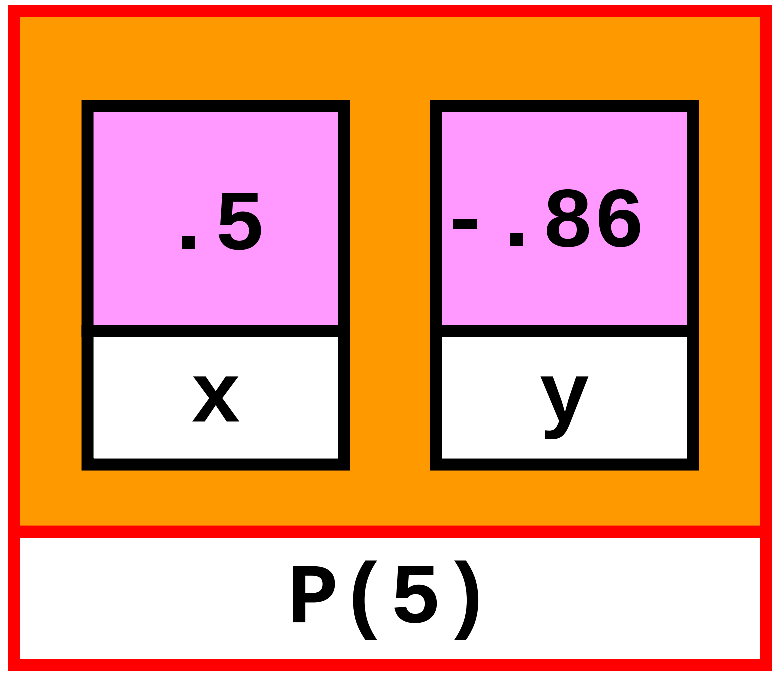




`P(4) = MakePoint(-.50,-.86)`

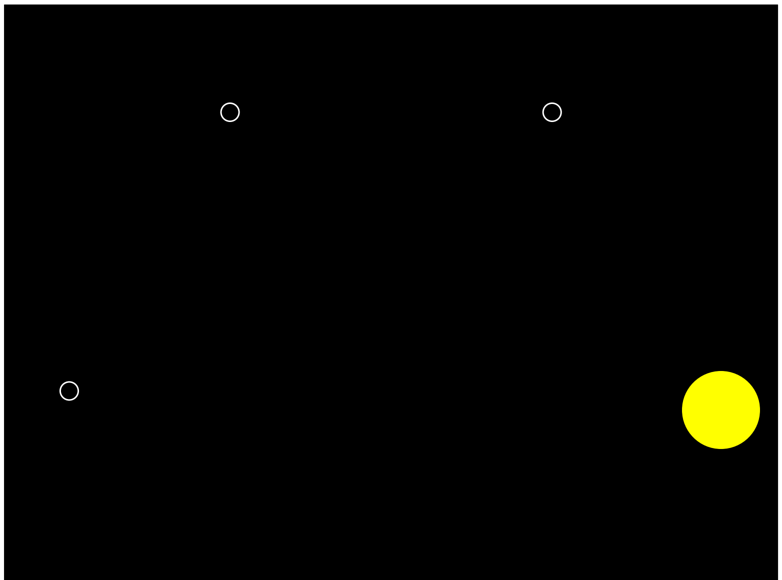
An Array of Points

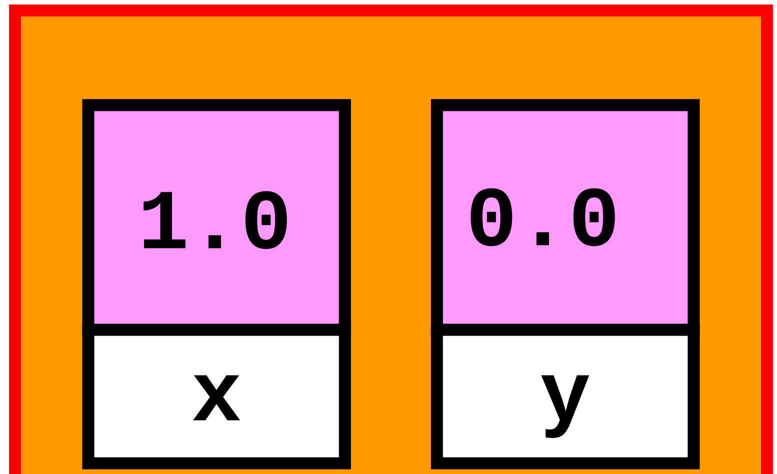





```
P(5) = MakePoint(.50,-.86)
```

An Array of Points






```
P(6) = MakePoint(1.0,0.0)
```

Function Returning An Array of Points

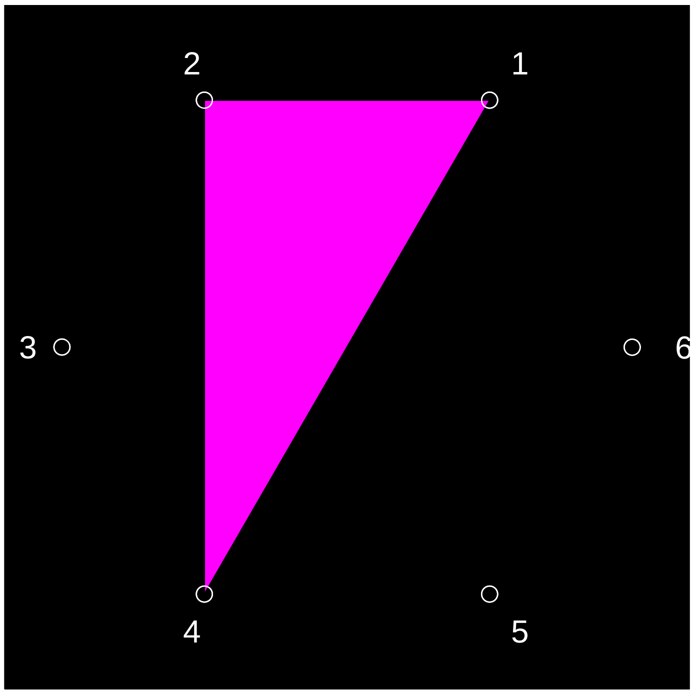
```
function P = CirclePoints(n)
% P is a structure array holding n points around a circle.

theta = 2*pi/n;
for k=1:n
   c = cos(theta*k);
   s = sin(theta*k);
   P(k) = MakePoint(c,s);
end
```

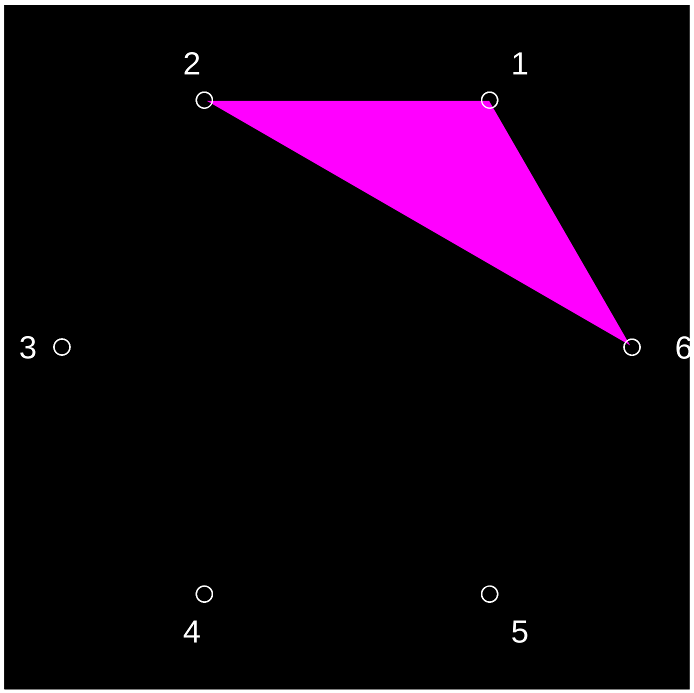
Example Problem

- Place n points uniformly around the unit circle
- Draw all possible triangles obtained by connecting these points 3-at-a-time

(i,j,k) = (1, 2, 4)



(i,j,k) = (1, 2, 6)

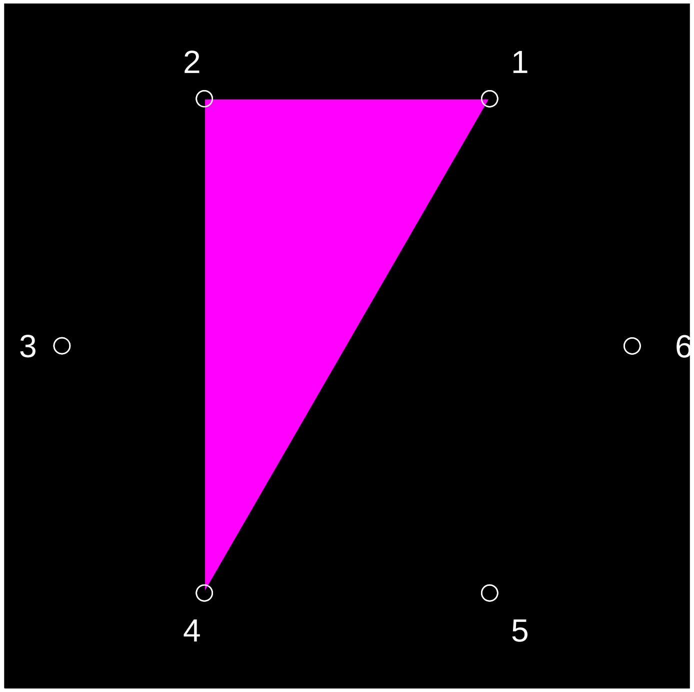


Will Need This…

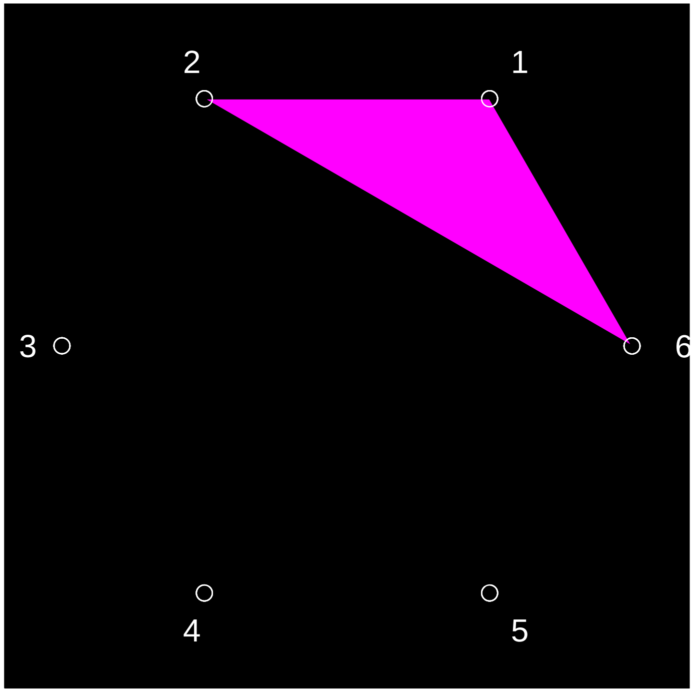
```
function DrawTriangle (P, Q, R, c)
% Draw c-colored triangle; triangle vertices are
% points P, Q, and R.

fill([P.x Q.x R.x P.x], [P.y Q.y R.y P.y], c)
```

(i , j , k) = (1 , 2 , 4)



(i , j , k) = (1 , 2 , 6)



These triangles are all the same:
(1,2,4), (1,4,2), (2,1,4), (2,4,1), (4,1,2), (4,2,1)

No!

```
for i=1:n
 for j=1:n
   for k=1:n
     Draw triangle with vertices P(i), P(j), and P(k)
   end
  end
end
```

i, j, and k should be different

Avoiding Duplicates: $i < j < k$

```
for i=1:n
  for j=i+1:n
    for k=j+1:n
        disp([i j k])
    end
  end
end
```

$i = 1$	$i = 2$	$i = 3$	$i = 4$
1 2 3	2 3 4	3 4 5	4 5 6
1 2 4	2 3 5	3 4 6	
1 2 5	2 3 6	3 5 6	
1 2 6	2 4 5		
1 3 4	2 4 6		
1 3 5	2 5 6		
1 3 6			
1 4 5			
1 4 6			
1 5 6			

Question Time

What is the 7th line of output:

```
for i=1:5
   for j=i+1:5
      x = 10*i + j
   end
end
```

A. 7 B. 21 C. 22 D. 23 E. Other

Triangle Solution!

```
for i=1:n
 for j=i+1:n
  for k=j+1:n
   DrawTriangle( P(i),P(j),P(k),'m')
   DrawPoints(P)
   pause
   DrawTriangle(P(i),P(j),P(k),'k')
  end
 end
end
```

Structures with Array Fields

- Let's develop a structure that can be use to represent a colored disk

- Four fields:

  xc: x-coordinate of center

  yc: y-coordinate of center

  r: radius

  c: rgb color vector

- Example:

```
D1 = struct('xc',1,'yc',2,'r',3,'c',[1 0 1])
D2 = struct('xc',4,'yc',0,'r',1,'c',[.2 .5 .3])
```

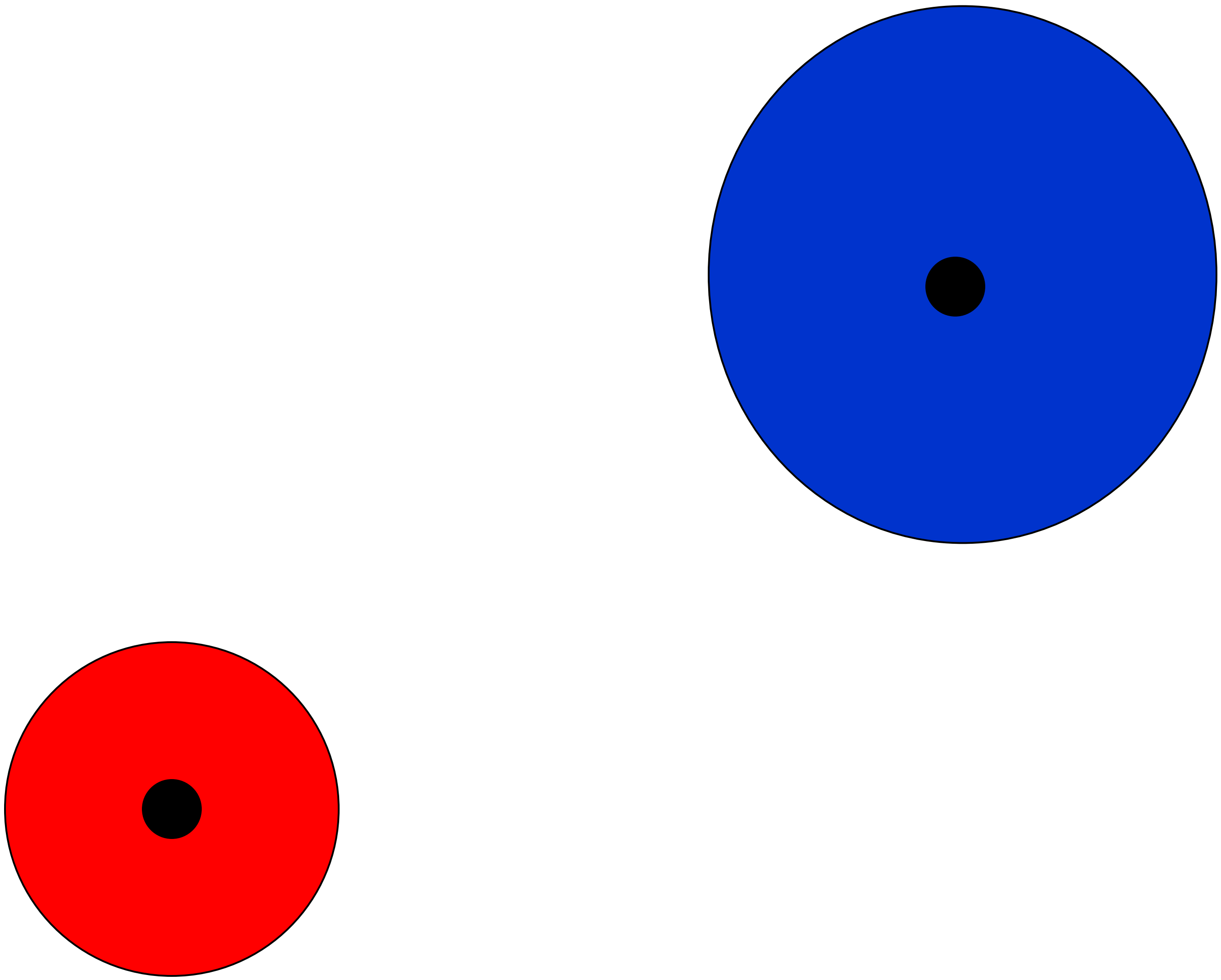
Problem

- Lets compute the "average" of D1 and D2:

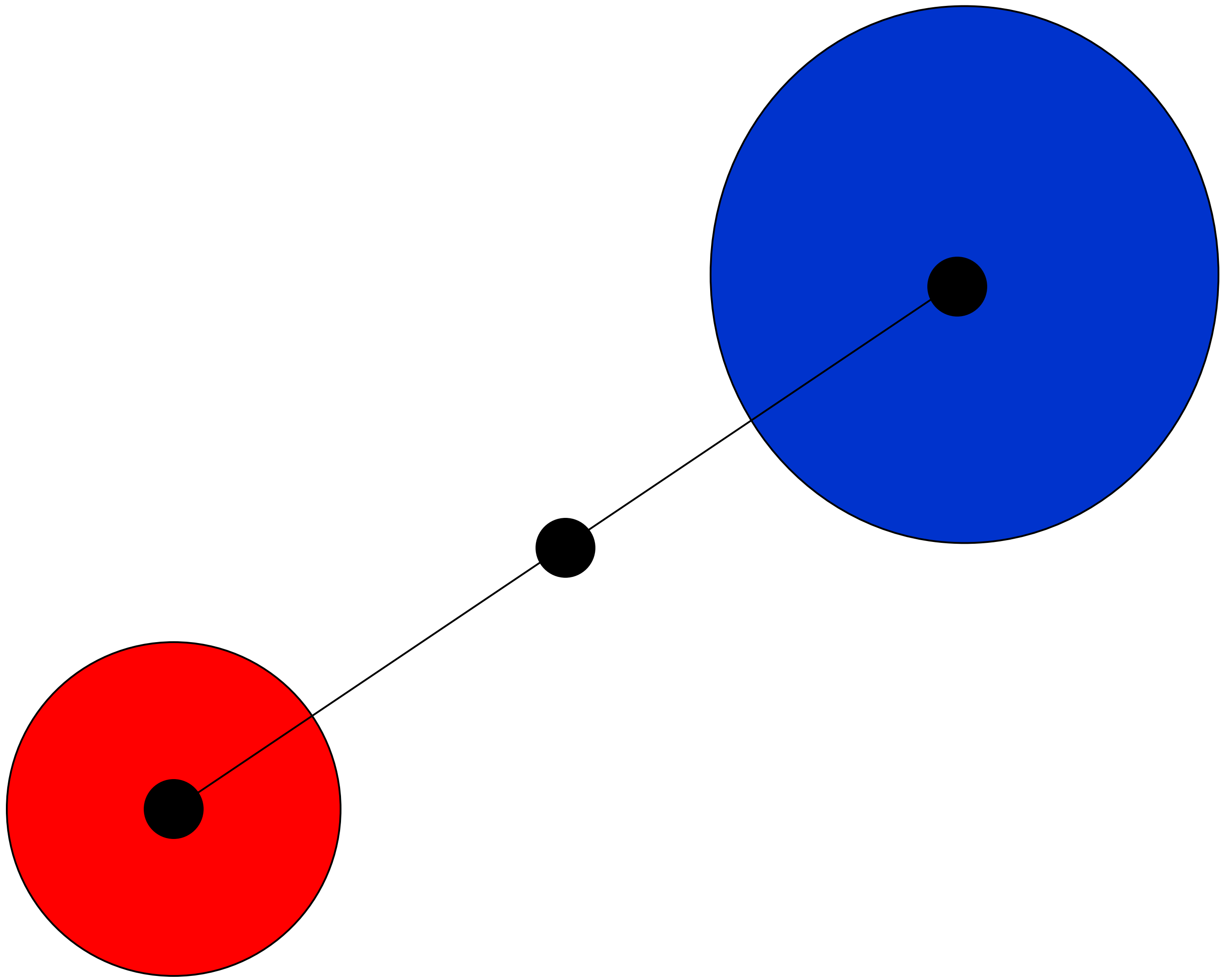
```
r  = (D1.r  + D2.r) /2;
xc = (D1.xc + D2.xc)/2
yc = (D1.yc + D2.yc)/2
c  = (D1.c  + D2.c) /2;

D = struct('xc',xc,'yc',yc,'r',r,'c',c);
```

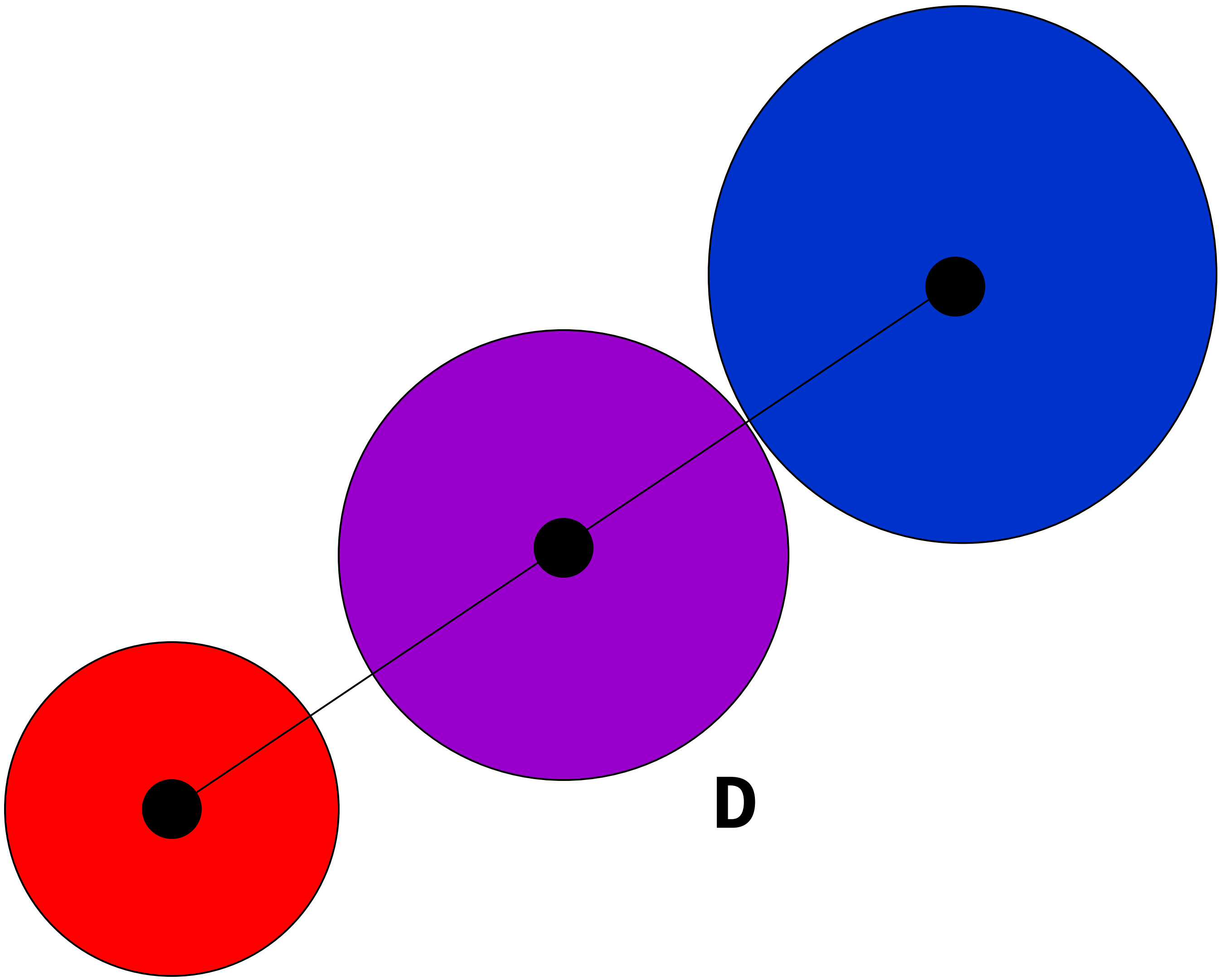
Example



Example



Example



A Structure's Field Can Hold a Structure

```
A = MakePoint(2,3)
B = MakePoint(4,5)
L = struct('P',A,'Q',B)
```

- This could be used to represent a line segment with endpoints P and Q, for instance

```
x = L.P.y        % Assigns 3 to x
```

Question Time

How do you set variable g to the green-color component of disk D?

```
D = struct('xc', xc, 'yc', yc, 'r', r, 'c', c);
```

A. g = D.g;
B. g = D.c.g;
C. g = D.c(2);
E. Other