L21. More on 2D Arrays

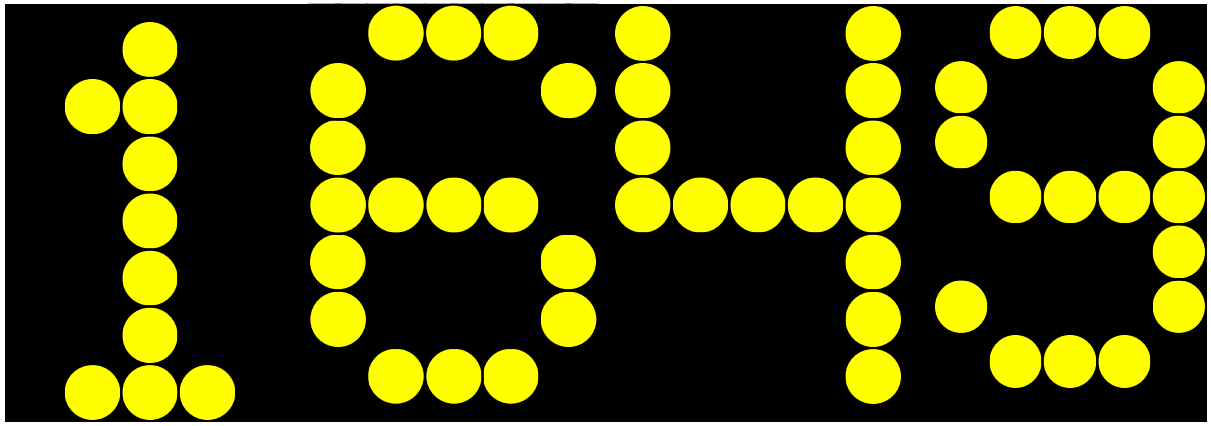
And their connections to

Cell arrays

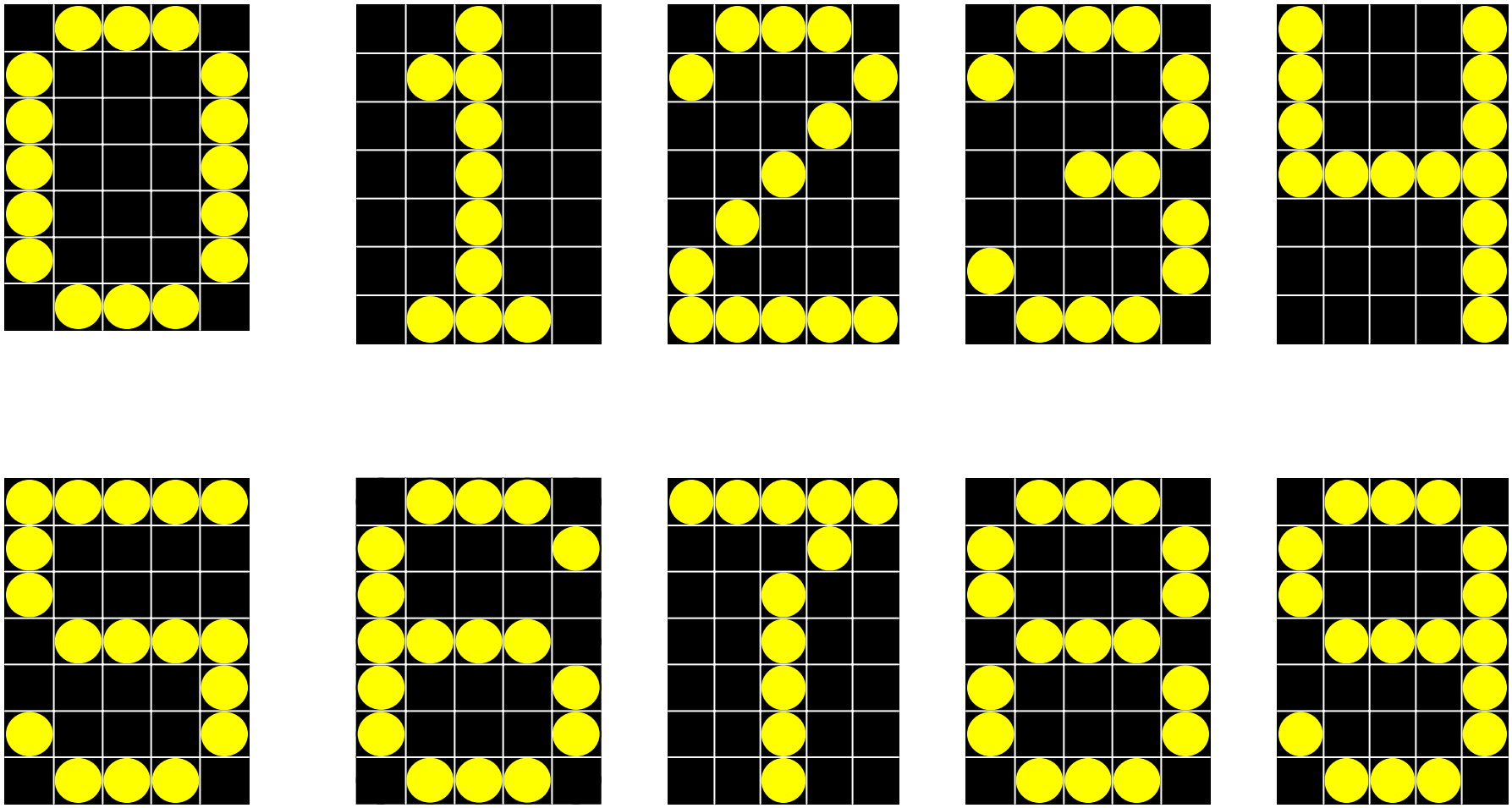
Structure arrays

Character arrays

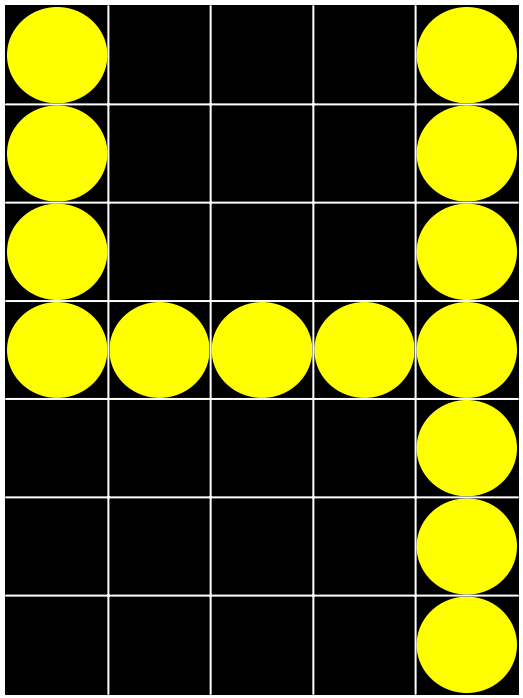
Application: Digital Displays



7-by-5 "Dot Matrices"



A "Bit Map" For Each Digit



A "light" is either on or off.

A 7-by-5 matrix of zeros and ones can "tell the whole story".

Look at Computations with These Bitmaps

First order of business:

Store the 10 bitmaps

Design Decisions

How do we package a particular digit?

numerical array or character array

How do we package the collection of digits?

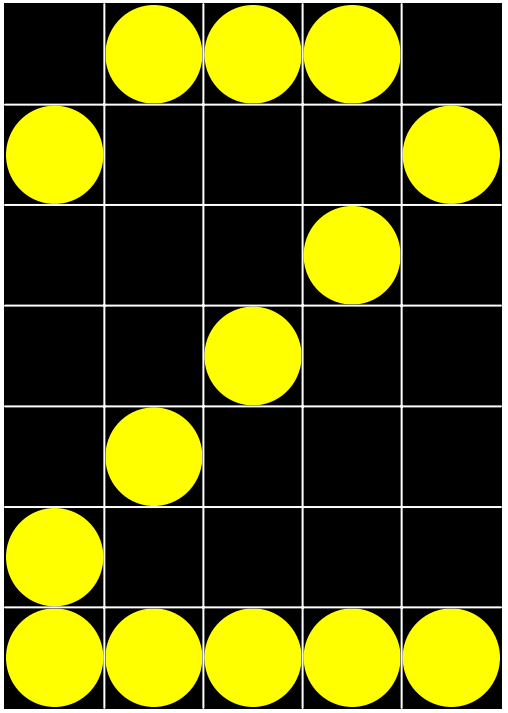
cell array or structure array

We look at the 4 possibilities.

Storing a Single Bitmap

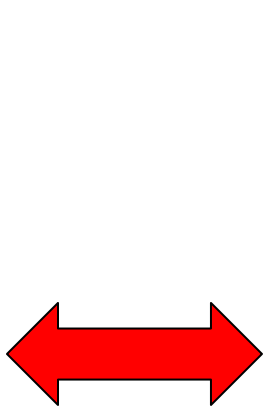
Can Use a Numerical Array For Each Digit

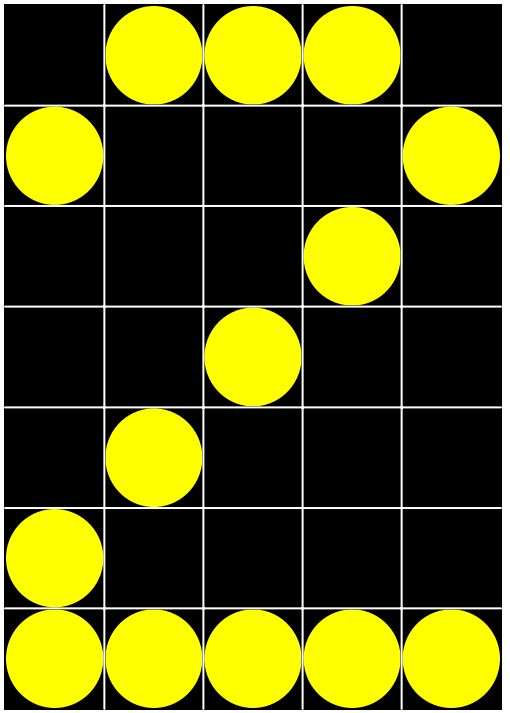
```
[ 0 1 1 1 0;…
  1 0 0 0 1;…
  0 0 0 1 0;…
  0 0 1 0 0;…
  0 1 0 0 0;…
  1 0 0 0 0;…
  1 1 1 1 1];
```



Can Use a Character Array For Each Digit

```
A  = [ '01110';…
       '10001';…
       '00010';…
       '00100';…
       '01000';…
       '10000';…
       '11111'];
```

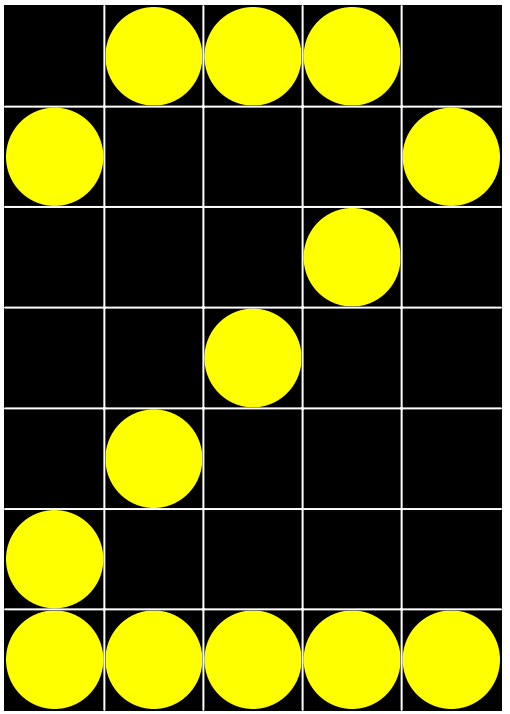




Storing the 10 Bitmaps in a Cell Array

Can Use a Cell Array this way

```
M = [ 0 1 1 1 0;…
      1 0 0 0 1;…
      0 0 0 1 0;…
      0 0 1 0 0;…
      0 1 0 0 0;…
      1 0 0 0 0;…
      1 1 1 1 1];
```



```
D{2} = M;
```

Here a cell is a numerical matrix

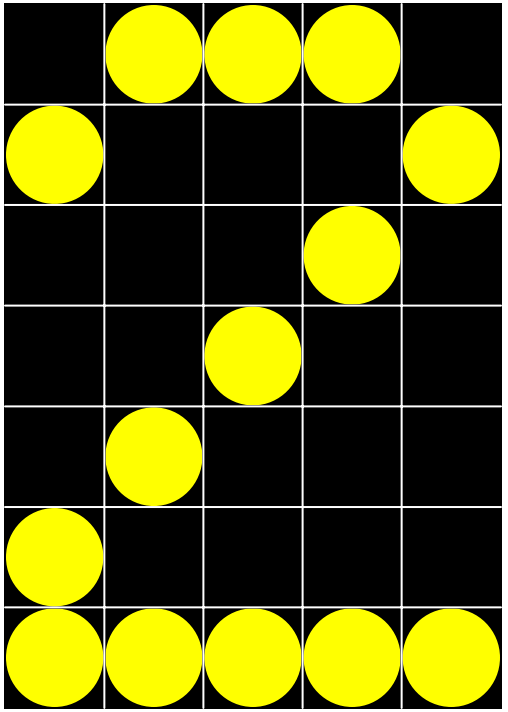
With D{1},…,D{10} Set Up:

```
M = D{k};
if M(4,3)==1
    disp('Middle Light is On')
end
```

Here k is initialized and satisfies 1<=k<=10

Or Can Use Cell Array this way…

```
M = [‘01110’;…
     ‘10001’;…
     ‘00010’;…
     ‘00100’;…
     ‘01000’;…
     ‘10000’;…
     ‘11111’];
```



```
D{2} = M;
```

Here a cell is a char array

With D{1},…,D{10} Set Up:

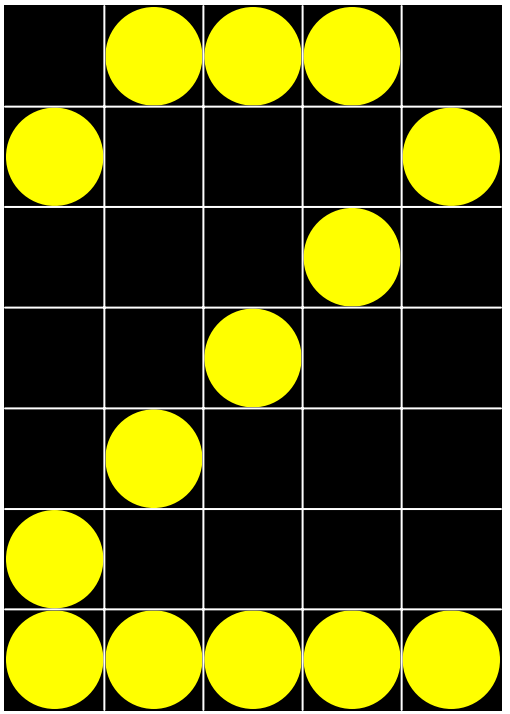
```
M = D{k};
if strcmp(M(4,3),'1')
    disp('Middle Light is On')
end
```

Here k is initialized and satisfies 1<=k<=10

Storing the 10 Bitmaps in a Structure Array

Can Use a Struct Array Like This

```
M = [ 0 1 1 1 0;…
      1 0 0 0 1;…
      0 0 0 1 0;…
      0 0 1 0 0;…
      0 1 0 0 0;…
      1 0 0 0 0;…
      1 1 1 1 1];
```



```
D(2) = struct('mat',M);
```

Here the sole field is a matrix

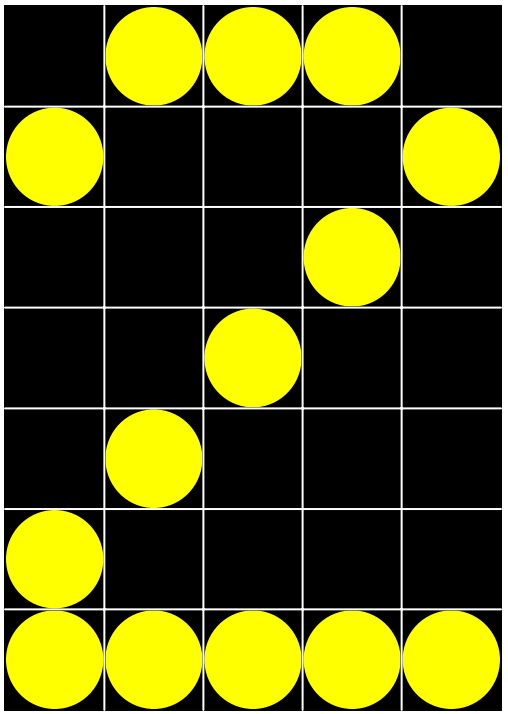
With D(1),…,D(10) Set Up:

```
M = D(k).mat;
if M(4,3)==1
     disp('Middle Light is On')
end
```

Here k is initialized and satisfies 1<=k<=10

Or a Struct Array Like This…

```
M = [ '01110';…
      '10001';…
      '00010';…
      '00100';…
      '01000';…
      '10000';…
      '11111'];
```



```
D(2) = struct('mat',M);
```

Here the sole field is a char array

With D(1),…,D(10) Set Up:

```
M = D(k).mat
if strcmp(M(4,3),'1')
    disp('Middle Light is On')
end
```

Here k is initialized and satisfies 1<=k<=10

Choice for Storing the Bit Maps

Cell array better than struct array

No point in having a structure with one field.

Numerical array better than char array

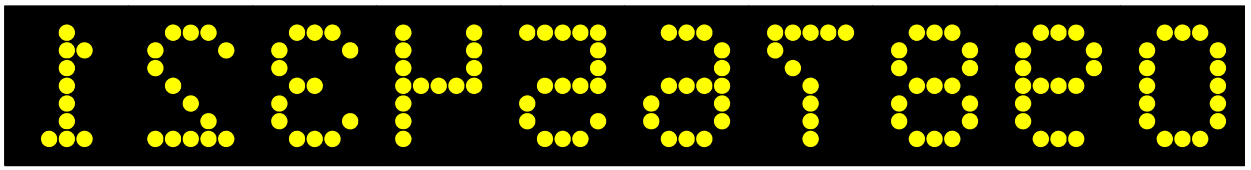
Plan on doing numerical computations with the bit arrays. Char arrays not handy

Assume Availability of This

```
function D = TheDigits
D = cell(10,1);
D{1}  = [0 0 1 0 0;...
         0 1 1 0 0;...
         0 0 1 0 0;...
         0 0 1 0 0;...
         0 0 1 0 0;...
         0 0 1 0 0;...
         0 1 1 1 0];
etc
```

Problem

Produce a cell array of "reverse" digits



Reversing Column Order

Suppose *A* has 5 columns. If

```
B(:,1) = A(:,5)
B(:,2) = A(:,4)
B(:,3) = A(:,3)
B(:,4) = A(:,2)
B(:,5) = A(:,1)
```

then B is *A* with its cols reversed.

```
B(:,k) = A(:,6-k)
```

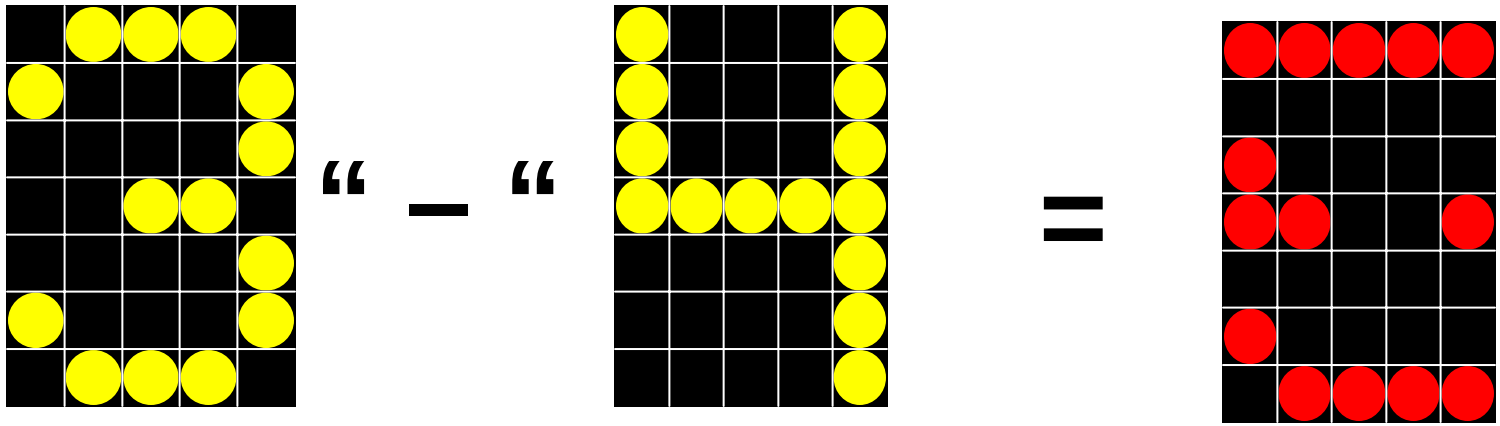
A Function to Do the Job

```
function B = ReverseCol(A)
[p,q] = size(A);
B = zeros(p,q);
for k=1:q
    B(:,k) = A(:,q-k+1);
end
```

A Cell Array of Reversed Digits

```
D = TheDigits;
revD = cell(10,1)
for k=1:10
    M = D{k};
    revM = ReverseCol(M);
    revD{k} = revM;
end
```

The Difference Between Two Bit Maps



A B C

```
C(i,j) = abs( A(i,j) - B(i,j) )
```

```
  function C = Difference(A,B)
% A and B are p-by-q arrays.
% C is a p-by-q array with
% C(i,j) = abs(A(i,j)-B(i,j))

  [p,q] = size(A);
  C = zeros(p,q);
  for i=1:p
      for j=1:q
          C(i,j) =  abs(A(i,j) - B(i,j));
      end
  end
```

Problem

100000 random digits are displayed in succession on a 7-by-5

How often does each of the 35 "bulbs" go on and off?

Digression: 2D Array Ops

```
>> A = [1 2;3 4];
>> B = [10 20; 30  40];
>> C = A + B
C =
        11   22
        33   44
```

Adding Up The Changes

```
D = TheDigits;
Count = zeros(7,5);
n = 10000;
for k=1:n
    i1 = ceil(10*rand);
    i2 = ceil(10*rand);
% M(i,j) = 1 if the two bitmaps disagree in
% position (i,j).
    M =  Difference(D{i1},D{i2});
    Count = Count + M;
end
```

Results

41979	31670	17754	31670	41979
31933	17936	17936	0	41913
48081	0	17936	17991	47770
48032	50078	31970	41836	41786
41818	0	41986	0	41986
49871	18011	31933	0	41986
18011	31707	17754	31707	31841