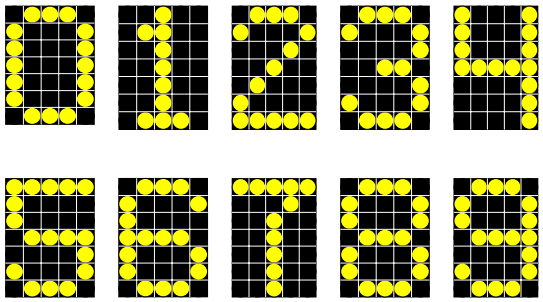
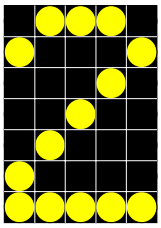
- Previous Lecture:
  - Working with large data files
  - Built-in `sort` function
- Today's Lecture:
  - Review matrix, cell array, structure array
- Announcement:
  - P5 due 4/11 at 3pm
  - Prelim 3 next Tues. We must know by now if you have an exam conflict.

7-by-5 "dot matrices"



Can use a cell array to keep the 10 bitmaps

```
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;
```

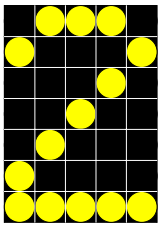
Each cell of cell array **D** is a numerical matrix.

With D{1},…,D{10} set up as a cell array of numerical matrices, can do computation as follows:

```
% given 1<=k<=10
M = D{k};
if M(4,3)==1
    disp('Middle light is on')
end
```


Still using a cell array to keep the 10 bitmaps

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



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

Each cell of cell array **D** is a matrix of characters.

With D{1},…,D{10} set up as a cell array of character matrices, can do computation as follows:

```
% given 1<=k<=10
M = D{k};
if strcmp(M(4,3),'1')
    disp('Middle light is on')
end
```

Using a structure array to keep the 10 bitmaps…
The k-th component of array D encodes digit k.

```
% Example for the digit 2
A = [ 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('matrix', A);
```

Which fragment on the right is correct given `1<=k<=10`?

A
```
M = D(k);
if M(4,3)==1
    disp('Middle light on')
end
```

B
```
M = D(k).matrix;
if M(4,3)==1
    disp('Middle light on')
end
```

C
```
M = D{k};
if M(4,3)==1
    disp('Middle light on')
end
```

D
```
M = D{k}.matrix;
if M(4,3)==1
    disp('Middle light on')
end
```

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 maps—char arrays not handy


```
function B = reverseCol(A)
% B is a matrix obtained by reversing
% the order of the columns in matrix A

[nr, nc]= size(A);
B= zeros(nr,nc);
for k= 1:nc
    B(:,k) = A(:,nc-k+1);
end
```



```
function revD = reverseDigits
% revD is a 10-by-1 cell array.
% revD{k} is the reversed 7-by-5 bitmap
% of digit k. revD{10} encodes 0.

D= TheDigits;
revD= cell(10,1);
```

```
function showNumber(n)
% Digital display of integer n, n>0

hold on; axis equal off

% Convert n to a vector of digits




% Display the digits in v
D = TheDigits;  % D{k} is matrix encoding digit k
```

```
% A and B have same size
[nr,nc]= size(A);
B= zeros(nr,nc);
for r= 1:nr
   for c= 1:nc
      C(r,c)= abs(A(r,c)-B(r,c));
   end
end
```

```
% A and B have same size
C= abs(A-B);
```

`C` is a 0-1 matrix where `1` indicates that `A(i,j)` and `B(i,j)` are *different*.