BE- Discussion #5, 2020-10-02

GOOD LUCK ON THE EXAM!

Announcements

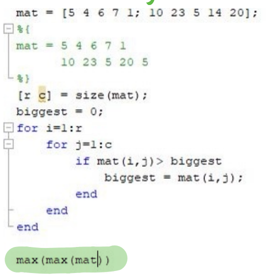
- HW#2 will be released this weekend
- Next Wednesday, October 7th is the last day to drop a course without a 'W"
- Exam 1 today $4^{30}-6^{00}$ pm EDT
  - to ask questions, you can either join the Zoom meeting and raise your hand, or you can email (list will be given on the exam instructions)

Review Quiz #2

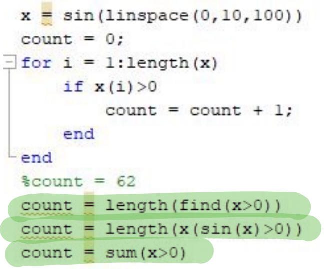
Review of Material

- Matrix multiplication
- vectorizing code
- load/save
- while loop

Vectorizing Code



```
mat = [5 4 6 7 1; 10 23 5 14 20];
%{
mat = 5 4 6 7 1
      10 23 5 20 5
%}
[r c] = size(mat);
biggest = 0;
for i=1:r
    for j=1:c
        if mat(i,j)> biggest
            biggest = mat(i,j);
        end
    end
end

max(max(mat))
```

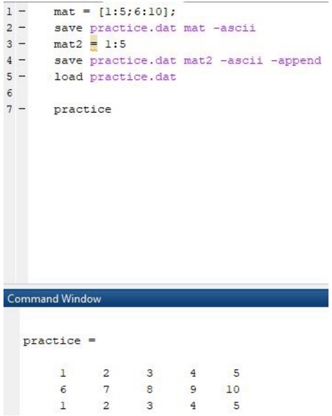


```
x = sin(linspace(0,10,100))
count = 0;
for i = 1:length(x)
    if x(i)>0
        count = count + 1;
    end
end
%count = 62
count = length(find(x>0))
count = length(x(sin(x)>0))
count = sum(x>0)
```

*checking to see how many positive numbers there are

*multiple ways to accomplish this!

load/save



```
mat = [1:5;6:10];
save practice.dat mat -ascii
mat2 = 1:5
save practice.dat mat2 -ascii -append
load practice.dat

practice
```

Command Window

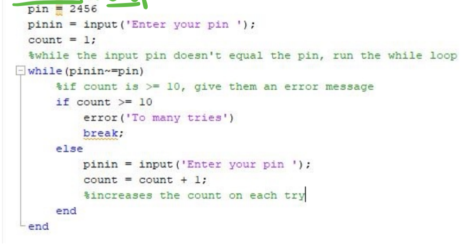
```
practice =

     1     2     3     4     5
     6     7     8     9    10
     1     2     3     4     5
```

"Un-vectorizing" code

newvec = diff(vec)

```
for i = 1:length(vec)-1
    newvec(i) = vec(i+1) - vec(i);
end
```

While loop



```
pin = 2456
pinin = input('Enter your pin ');
count = 1;
%while the input pin doesn't equal the pin, run the while loop
while(pinin~=pin)
    %if count is >= 10, give them an error message
    if count >= 10
        error('To many tries')
        break;
    else
        pinin = input('Enter your pin ');
        count = count + 1;
        %increases the count on each try
    end
end
```

*remember the atm/pin # exercise from the GPP!

Matrix Multiplication

mat1 =

$$\underset{3 \times 3}{\begin{matrix}1 & 2 & 3\\4 & 5 & 6\\7 & 8 & 9\end{matrix}}$$

mat2 =

$$\underset{2 \times 3}{\begin{matrix}1 & 2 & 3\\3 & 2 & 1\end{matrix}}$$

mat3 =

$$\underset{2 \times 3}{\begin{matrix}4 & 2 & 0\\2 & 6 & 8\end{matrix}}$$

```
>> mat1 * mat2
```

error, dimensions don't match

```
>> mat 2 * mat 1
```

$$\underset{2 \times 3}{\begin{bmatrix}1 & 2 & 3\\3 & 2 & 1\end{bmatrix}}\underset{3 \times 3}{\begin{bmatrix}1 & 2 & 3\\4 & 5 & 6\\7 & 8 & 9\end{bmatrix}}$$

3 = 3, inner dimensions match, VALID!

outer dimensions determine final size which is 2x3

(1)(1) + (2)(4) + (3)(7) = 30   (1)(2) + (2)(5) + (3)(8) = 36   (1)(3) + (2)(6) + (3)(9) = 42

(3)(1) + (2)(4) + (1)(7) = 18   (3)(2) + (2)(5) + (1)(8) = 24   (3)(3) + (2)(6) + (1)(9) = 30

$$\boxed{\begin{matrix}30 & 36 & 42\\18 & 24 & 30\end{matrix}}$$

```
>> mat2 * mat3
```

error, dimensions don't match

```
>> mat2 .* mat3
```

$$\begin{bmatrix}1 & 2 & 3\\3 & 2 & 1\end{bmatrix} .* \begin{bmatrix}4 & 2 & 0\\2 & 6 & 8\end{bmatrix} \Rightarrow \begin{matrix}(1)(4) = 4 & (2)(2) = 4 & (3)(0) = 0\\(3)(2) = 6 & (2)(6) = 12 & (1)(8) = 8\end{matrix} \Rightarrow \boxed{\begin{matrix}4 & 4 & 0\\6 & 12 & 8\end{matrix}}$$

```
>> 3 * mat2
```

$$\boxed{\begin{matrix}3 & 6 & 9\\9 & 6 & 3\end{matrix}}$$

```
>> mat2 .* 3
```

$$\boxed{\begin{matrix}3 & 6 & 9\\9 & 6 & 3\end{matrix}}$$

Example function

```
function area = areaCircle(radius)
% Calculates the area of a circle
% radius is the input argument
area = pi * radius.^2;
% area is the returned variable
end
```

ex. of calling:

```
>> areaCircle(5)
    ans =
          78.5398
```

*NOTE: reserved word "function" and don't ever forget the 'end'!