```
%HW 2.mat
%Patrick Utz, 1/19/18, 2.1.1
%For each of the following math expressions, use MATLAB to calculate the result for the given
variable values. To be more general, each math expression should be defined using the variables.
Math Expression: 3cos(2πωt+50°) when ω=1000rad/second and t =10 second
%w = angular velocity, t = time, answer = final result using radians to calculate

clear, clc
>> w = 1000
w =
      1000
>> t = 10
t =
   10
>> answer = 3*( cos( (2*pi*w*t) + 0.872665) )


answer =
   1.9284
```

%HW 2.mat
%Patrick Utz, 1/19/18, 2.1.2
%For each of the following math expressions, use MATLAB to calculate the result for the given variable values. To be more general, each math expression should be defined using the variables.
Math Expression: $10 \log_{20}(\left|\frac{V_0}{V_i}\right|)$, when V_0=100 volts and V_i = 1 volt
%vnot = V_0, vinit =V_i, answer = final result

```
clear, clc
>> vnot = 100
vnot =
  100
>> vinit = 1
vinit =
    1
>> answer = 10*( (log(abs(vnot/vinit))) / (log(20)) )


answer =
  15.3724
```

```
%HW 2.mat
%Patrick Utz, 1/19/18, 2.1.3
%For each of the following math expressions, use MATLAB to calculate the result for the given
variable values. To be more general, each math expression should be defined using the variables.
```

Math Expression: $e^{-t/RC} - 1$, when t=1 second; R = 1000 ohms and C = 0.001Farad

```
%t = time in seconds, r = R in ohms, c = C in Farads, answer = final result

clear, clc
>> t = 1
t =
    1
>> r = 1000
r =
       1000
>> c = .001
c =
   1.0000e-03
>> answer = ( exp( (-t)/r*c ) ) - 1


answer =
  -1.0000e-06
```

%HW 2.mat
%Patrick Utz, 1/19/18, 2.1.4
%For each of the following math expressions, use MATLAB to calculate the result for the given variable values. To be more general, each math expression should be defined using the variables.
Math Expression: $\sqrt[3]{a^{2.2^3} + b^{\frac{2}{3}}}$, when a = 2 and b = 3
%a = a, b = b, answer = final result

```
clear, clc
>> a = 2
a =
    2
>> b = 3
b =
    3
>> answer = nthroot( (a^(2.2^3) + b^(2/3)), 3 )


answer =
   11.7123
```

%HW 2.mat
%Patrick Utz, 1/19/18, 2.2
%Create each of the following row vectors using two methods: the colon operator and the **linspace** function:

1) [-5 -4 -3 -2 -1]
2) [10 8 6 4]

%v = row vector 1, m = row vector 2

1) clear, clc

```
>> v = -5:1:-1

v =
   -5    -4    -3    -2    -1

>> v = linspace(-5,-1,5)

v =
   -5    -4    -3    -2    -1
```

2) clear, clc

```
>> m = 10:-2:4

m =
   10     8     6     4

>> m = linspace(10,4,4)

m =
   10     8     6     4
```

%HW 2.mat
%Patrick Utz, 1/19/18, 2.3
%Write an expression that refers to only the odd-numbered elements in a vector, regardless of the length of the vector. Write another expression that refers to only the even-numbered elements in a vector, regardless of the length of the vector. Test your expressions on the following two vectors:

1) [4 5 6 7 8 9 10 11]
2) [0 $\frac{\pi}{2}$ π $\frac{3\pi}{2}$ 2π]

%v = row vector 1, m = row vector 2, size = the number of elements in the vector

1) clear, clc

```
>> v = 4:11;              Even
size = numel(v);
>> v(1:2:size) = [];
>> v
v =
    5    7    9    11

>> v = 4:11;              Odd
size = numel(v);
>> v(2:2:size) = [];
>> v
v =
    4    6    8    10
```

2) clear, clc **Even**

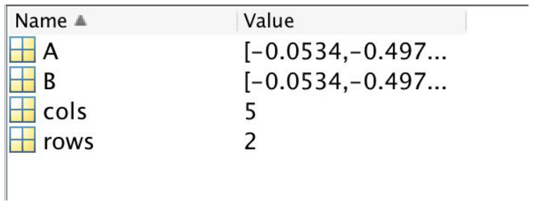
```
>> m = 0:(pi/2):(2*pi);
>> size = numel(m);
>> m(1:2:size) = [];
>> m
m =
   1.5708    4.7124

>> m = 0:(pi/2):(2*pi);          Odd
>> size = numel(m);
>> m(2:2:size) = [];
>> m
m =
        0    3.1416    6.2832
```

```
%HW 2.mat
%Patrick Utz, 1/19/18, 2.4
%Create a vector x which consists of 20 equally spaced points in the range from –π
to +π. Create a y vector which is sin(x)
%x = vector x, y = vector y which is sin(x)

clear, clc

>> x = linspace(-pi,pi,20);
>> y = sin(x);
>> x

x =
  Columns 1 through 11
   -3.1416   -2.8109   -2.4802   -2.1495   -1.8188   -1.4881   -1.1574   -0.8267   -0.4960   -0.1653
0.1653
  Columns 12 through 20
    0.4960    0.8267    1.1574    1.4881    1.8188    2.1495    2.4802    2.8109    3.1416

>> y

y =
  Columns 1 through 10
   -0.0000   -0.3247   -0.6142   -0.8372   -0.9694   -0.9966   -0.9158   -0.7357   -0.4759   -0.1646
  Columns 11 through 20
    0.1646    0.4759    0.7357    0.9158    0.9966    0.9694    0.8372    0.6142    0.3247    0.0000
```

%HW 2.mat
%Patrick Utz, 1/19/18, 2.5
%Create a variable *rows* that is a random integer in the inclusive range from 1 to 5; create a variable *cols* that is a random integer in the inclusive range from 1 to 5; then create a matrix in a variable named A of random real numbers in the inclusive range from -1 to +1 with the dimensions given by the values of rows and cols. Then reverse the order of every odd-numbered row and store the resulting matrix in a new variable named B. Your statement should be general. For example: if rows = 5; and cols = 5
Test your solution using two different A matrices, one with an odd-number of rows and the other with an even-number of rows. Use the same set of commands for both cases.
%x = vector x, y = vector y which is sin(x)

clear, clc

With Even # of Rows



```
Name ▲        Value
A             [-0.0534,-0.497...
B             [-0.0534,-0.497...
cols          5
rows          2
```

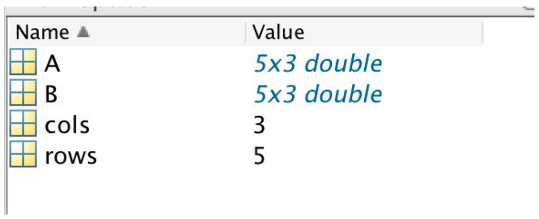
```
>> rows = randi(5);
>> clear
>> rows = randi(5);
>> cols = randi(5);
>> A = -1 + (1+1).*rand(rows,cols)
A =
   -0.4914   -0.5130   -0.3000   -0.4978   -0.0534
    0.6286    0.8585   -0.6068    0.2321   -0.2967
>> A(1:2:end,:)=fliplr(A(1:2:end,:))
A =
   -0.0534   -0.4978   -0.3000   -0.5130   -0.4914
    0.6286    0.8585   -0.6068    0.2321   -0.2967
>> B = A

B =
   -0.0534   -0.4978   -0.3000   -0.5130   -0.4914
    0.6286    0.8585   -0.6068    0.2321   -0.2967
```

Cont. —>

2.5 cont.

With Odd # of Rows

```
>> rows = randi(5);
>> cols = randi(5);
>> A = -1 + (1+1).*rand(rows,cols)
A =
    0.0994   -0.2391    0.5583
    0.8344    0.1356    0.8680
   -0.4283   -0.8483   -0.7402
    0.5144   -0.8921    0.1376
    0.5075    0.0616   -0.0612
>> A(1:2:end,:)=fliplr(A(1:2:end,:))
A =
    0.5583   -0.2391    0.0994
    0.8344    0.1356    0.8680
   -0.7402   -0.8483   -0.4283
    0.5144   -0.8921    0.1376
   -0.0612    0.0616    0.5075
>> B = A

B =
    0.5583   -0.2391    0.0994
    0.8344    0.1356    0.8680
   -0.7402   -0.8483   -0.4283
    0.5144   -0.8921    0.1376
   -0.0612    0.0616    0.5075
```



Name	Value
A	5x3 double
B	5x3 double
cols	3
rows	5