```
% class 2


a = 2+2
     4
a = 2>2
     0
a =23
y=21
z = a+y
    44

z = a+b
Undefined function or variable 'b'.

a = a + 2

a = a + a*0.1  % increase value of a by 10%
a = a*1.1
a = a + a/10


v = [1 2 3]
v = [1, 2, 3]


whos    %display variables in workspace
  Name      Size            Bytes  Class     Attributes

  a         1x1                 8  double
  ans       1x1                 8  double
  v         1x3                24  double
  y         1x1                 8  double
  z         1x1                 8  double


v = [5 6 7 8]
     5     6     7     8
v = 5:8                 % create vector starting with 5 and ending with 8
     5     6     7     8
v = 5:80
v = 5:10:80    % create vector with increments of 10
     5    15    25    35    45    55    65    75

Exercise: Make v = [-6 -8 -10 -12]
      v = -6:-2:-12
      v = -1*[6:2:12]

a = [2 3 4]

     2     3     4

a = [2 3 4]'   % transposed vector

     2
     3
     4
```

```
size( a )

     3     1

size( a' )
      1    3



length(a)  % similar to ‘size’ function but returns only longer dimension

     3

length(a')

     3

a = [2 3 4]
a(2)       %access 2nd element of vector ‘a’
     3

a(3)
     4

v = [ 5 6 7 8]
     5     6     7     8

v(4)

     8
```

Exercise: get 3 and 4th element of v

```
      v( [3 4])


vArr = [ 5 6 7 8; 2 3 4 5 ]  % create array

     5     6     7     8
     2     3     4     5

vArr (2,2)
     3

vArr (2,1)  % access element in 2nd row and 1st column of array vArr
     2

size( vArr )

     2     4


Arr = [ 5 6 7 8; 2 3 4 5 ]

Arr =

     5     6     7     8
     2     3     4     5


Arr(2,2) = 100   % change value of single element in Arr

     5     6     7     8
     2   100     4     5

Arr =  Arr -2

     3     4     5     6
     0    98     2     3




v = [1 2]^2
Error using  ^
Inputs must be a scalar and a square matrix.
To compute elementwise POWER, use POWER (.^)
instead.

v = [1 2].^2

     1     4

v1 = [1 2]

v2 = [2 3]


v1*v2
Error using  *
Inner matrix dimensions must agree.

v1*v2'  % so-called inner product in algebra

     8

v1.*v2   % multiplication element-by-element

     2     6


Arr1 = [1 2; 3 4]

     1     2
     3     4

Arr2 =[ 3 4 ; 5 6]

     3     4
     5     6

Arr1.*Arr2

     3     8
```

```
    15    24

Arr1*Arr2

    13    16
    29    36




Arr

Arr =

     3     4     5     6
     0    98     2     3

Arr( 1, 2:4)  % access multiple elements

     4     5     6

Arr( 1:2, 2:4)

     4     5     6
    98     2     3

Arr( 1, : )  %  access all elements in 1st row

     3     4     5     6

Arr( 1, 2:end )  % access elements from 2nd till last

     4     5     6

v = [2 4 7 ]

v(end)
     7


v = [  2     4     7 ]

find( v > 3 )  % find indices of elements in v bigger than 3
     2     3

find( v < 3 )

     1

find( v < -3 )

   Empty matrix: 1-by-0


Arr
     3     4     5     6
     0    98     2     3
```

```
find(Arr>3)

     3
     4
     5
     7


[x y] = find( Arr>3 )   % return row and column index of elements >3

x =

     1
     2
     1
     1


y =

     2
     2
     3
     4

xx = find(Arr(2,:)>3)    % search only in 2nd row

     2

Arr(2, xx)    % display value of that element which  was >3

    98

xx = find( Arr( 2,:) > 1 )

     2     3     4

Arr(2, xx)
    98     2     3
```

Exercise: a=[1 2]; b=[3 4], make from ab vector [a b] or array [a;b]

v=[a b]; v=[a; b];

```
5 == 3   % checking logical condition
  0
5 = 3
  |
Error: The expression to the left of the equals
sign is not a valid target for an assignment.

 v = [2 3 4]
  2   3   4
```

```
find( v == 4 )
    3
find( v >= 4 )
find( v <= 4 )
    1     2     3
find( v =< 4 )                    %  sign '=' must be after '<'
        |
Error: The expression to the left of the equals sign is not a valid target for an assignment.


v =[     2     3     4]

 find( v>2 & v<4 )  % AND
    2
 find( v>2 & v<=4 )
    2     3

x = find( v>2 & v<=4 )

    2     3


 v2 = [3 5 9 8 5 ]
    3     5     9     8     5

 x=find(v2==5)

find( v<2 | v>4 )  % OR

find( ( v>2 & v<5 ) | v>10 )
```