22000862

R. K. J. P. Kashmira

Octave Lab Practical Sheet- 02

```
octave:1> sqrt(98)
ans = 9.8995
octave:2> 98^0.5
ans = 9.8995
octave:3> 98^1/2
ans = 49
octave:4> 98^(1/2)
ans = 9.8995
octave:5>
```

```
octave:5> y = 49;
octave:6> floor(sqrt(y)) == sqrt(y)
ans = 1
```

```
octave:7> iskeyword("if")
ans = 1
octave:8> iskeyword("when")
ans = 0
octave:9> iskeyword("while")
ans = 1
octave:10> iskeyword("how")
ans = 0
octave:11> iskeyword("which")
ans = 0
octave:12> iskeyword("catch")
ans = 1
octave:13> iskeyword("try")
ans = 1
octave:14> iskeyword("until")
ans = 1
octave:15> iskeyword("spmd")
ans = 1
octave:16> iskeyword("spot")
ans = 0
octave:17> iskeyword("partfor")
ans = 0
octave:18> iskeyword("for")
ans = 1
octave:19> iskeyword("global")
ans = 1
octave:20> iskeyword("else")
ans = 1
octave:21> iskeyword("e")
ans = 0
octave:22> iskeyword("pi")
ans = 0
octave:23> iskeyword("__FINE__")
ans = 0
octave:24> iskeyword("__LINE__")
ans = 1
octave:25> iskeyword("break")
ans = 1
octave:26> iskeyword("broke")
ans = 0
octave:27> iskeyword("function")
ans = 1
```

```
octave:28> a = 5; b = [1, 2, 3];
octave:29> who
Variables visible from the current scope:

a    ans  b    y

octave:30> whos
Variables visible from the current scope:

variables in scope: top scope

  Attr   Name        Size                     Bytes  Class
  ====   ====        ====                     =====  =====
         a           1x1                          8  double
         ans         1x1                          1  logical
         b           1x3                         24  double
         y           1x1                          8  double

Total is 6 elements using 41 bytes
```

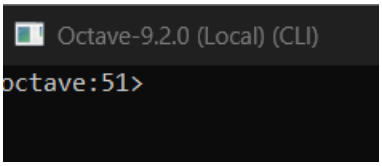
```
octave:31> a = 5;
octave:32> b = [1, 2; 3, 4];
octave:33> size(a)
ans =

   1   1

octave:34> size(b)
ans =

   2   2
```

```
octave:35> Inf / 5
ans = Inf
octave:36> Inf / 0
ans = Inf
octave:37> Inf / -5
ans = -Inf
octave:38> Inf / (3 + 4i)
ans =  Inf - Infi
octave:39>
```

```
octave:39> Inf / Inf
ans = NaN
octave:40> Inf^2
ans = Inf
octave:41> sqrt(Inf)
ans = Inf
octave:42> Inf + Inf
ans = Inf
octave:43> Inf * Inf
ans = Inf
octave:44> Inf - Inf
ans = NaN
octave:45> sqrt(-Inf)
ans =    0 + Infi
octave:46> Inf^Inf
ans = Inf
octave:47> Inf / i
ans =  NaN - Infi
octave:48>
```

16. What is the usage of the clc command? Type it on the command line and find out??

Clear the cli



17. What is the usage of the clear command? Type it on the command line and find out.

This command removes variables from the workspace

```
octave:51> 2+3
ans = 5
octave:52> clear
octave:53> a=5
a = 5
octave:54> clear
octave:55> a
error: 'a' undefined near line 1, column 1
octave:56>
```

```
octave:56> format short
octave:57> pi + e
ans = 5.8599
octave:58> format long
octave:59> pi + e
ans = 5.859874482048838
octave:60>
```

```
octave:60> X = [2, 3; 4, 1];
octave:61> A = [3, 4, 10; 70, 1, 30];
octave:62> Z = [2, 3; 50, 49; 0, 1];
octave:63> Y = [1, 0, 0; 0, 1, 0; 0, 0, 1];
octave:64> X'
ans =

   2   4
   3   1

octave:65> A'
ans =

    3   70
    4    1
   10   30

octave:66> Z'
ans =

    2   50    0
    3   49    1

octave:67> Y'
ans =

   1   0   0
   0   1   0
   0   0   1
```

```
octave:68> fliplr(X)
ans =

   3   2
   1   4

octave:69> fliplr(A)
ans =

   10    4    3
   30    1   70

octave:70> fliplr(Z)
ans =

    3    2
   49   50
    1    0

octave:71> fliplr(Y)
ans =

   0   0   1
   0   1   0
   1   0   0

octave:72>
```

```
ctave:72> flipud(X)
ns =

  4   1
  2   3

ctave:73> flipud(A)
ns =

  70    1   30
   3    4   10

ctave:74> flipud(Z)
ns =

   0    1
  50   49
   2    3

ctave:75> flipud(Y)
ns =

  0   0   1
  0   1   0
  1   0   0

ctave:76> x = 7;
ctave:77> y = x^2 - 6*x + 5
 = 12
ctave:78> linspace(5, 150, 11)
ns =

Columns 1 through 4:

  5.000000000000000e+00   1.950000000000000e+01   3.400000000000000e+01   4.850000000000000e+01

Columns 5 through 8:

  6.300000000000000e+01   7.750000000000000e+01   9.200000000000000e+01   1.065000000000000e+02

Columns 9 through 11:

  1.210000000000000e+02   1.355000000000000e+02   1.500000000000000e+02

ctave:79>
```

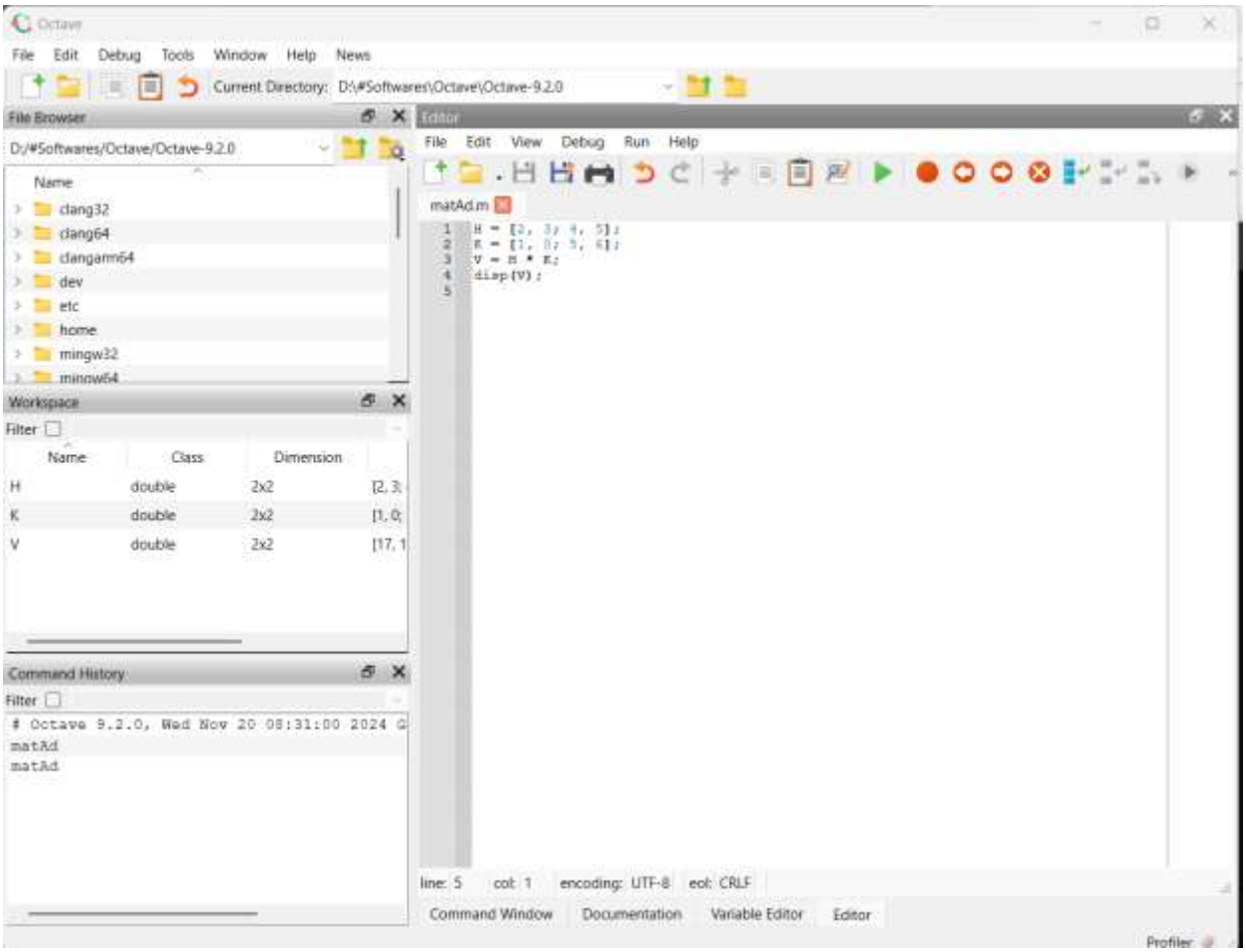
```
octave:79> logspace(1, 3, 7)
ans =

 Columns 1 through 4:

   1.000000000000000e+01   2.154434690031883e+01   4.641588833612777e+01   1.000000000000000e+02

 Columns 5 through 7:

   2.154434690031885e+02   4.641588833612777e+02   1.000000000000000e+03

octave:80> rand(3, 4)
ans =

   6.045960625855922e-01   5.261988429343434e-02   8.893559813865722e-01   6.056289444654972e-02
   4.288005507710142e-01   3.375961036612360e-01   2.082424912122454e-01   7.130838022472944e-01
   3.930834625759028e-01   8.333715092909288e-02   4.373068962416016e-01   7.772312107330277e-01

octave:81> rand(3, 4)
ans =

   0.589594353778886   0.730664566763906   0.187712302437022   0.634757324511140
   0.226581163397446   0.154242244112787   0.227922417797886   0.189788392533725
   0.778668784811294   0.623977866015728   0.859307743369499   0.650358538529928

octave:82> rand(3, 4)
ans =

   2.717277967487568e-01   9.762579216881917e-01   3.046675498063034e-02   3.481992791802729e-01
   1.786066959855537e-01   2.694813312779731e-01   4.091553288160610e-02   1.816356361926269e-01
   9.804840833039367e-01   2.179114221211126e-01   9.341550315392753e-01   3.217367948657002e-02

octave:83> rand(3, 4)
ans =

   7.043575628032381e-01   9.439578541758608e-01   6.955359102947630e-01   6.655794701134587e-01
   3.342419512114467e-01   7.766927345588377e-01   3.791519623088538e-01   4.906010189986898e-02
   4.530895411268639e-01   9.525179662574975e-01   8.433975067676114e-01   1.060737754986139e-01

octave:84> rand(3, 4)
ans =

   5.070146070007157e-01   5.814349430943389e-03   3.313425600417211e-01   6.014735401442273e-02
   2.765559697614719e-01   7.915184134416186e-01   3.588815537782140e-01   3.320831657892622e-01
   4.278765335781595e-01   1.298148164830158e-01   2.483699455820497e-01   9.446718736856590e-01

octave:85>
```

```
octave:89> R = rand(3, 4);
octave:90> R
R =

   3.661703059168316e-01   1.673598827165679e-01   5.609293070859531e-01   5.824018045980868e-01
   5.376491891042534e-01   7.836874500332928e-01   4.193987919445964e-01   9.486819330324225e-02
   5.880199224114169e-01   6.286494823719478e-01   3.405004735217083e-01   2.985292199435633e-01
```

```
     ^
octave:91> X = X + 25
X =

   27   28
   29   26

octave:92> X = X + 25
X =

   52   53
   54   51

octave:93> X = X + 25
X =

   77   78
   79   76

octave:94> X = X + 25
X =

   102   103
   104   101

octave:95> X = X + 25
X =

   127   128
   129   126

octave:96> X = X + 25
X =

   152   153
   154   151

octave:97> X = X + 25
X =

   177   178
   179   176

octave:98> X = X + 25
X =

   202   203
   204   201
```

```
octave:99> a = 2;
octave:100> z = 8;
octave:101> for i = 1:14
>     a = a + 1;
>     z = z + 2;
> end
octave:102> a
a = 16
octave:103> z
z = 36
octave:104>
```

```
Select Octave-9.2.0 (Local) (CLI)
octave:104> inv([2, 3; 4, 5])
ans =

  -2.500000000000000   1.500000000000000
   2.000000000000000  -1.000000000000000

octave:105> y = eye(12);
octave:106> y
y =

Diagonal Matrix

   1   0   0   0   0   0   0   0   0   0   0   0
   0   1   0   0   0   0   0   0   0   0   0   0
   0   0   1   0   0   0   0   0   0   0   0   0
   0   0   0   1   0   0   0   0   0   0   0   0
   0   0   0   0   1   0   0   0   0   0   0   0
   0   0   0   0   0   1   0   0   0   0   0   0
   0   0   0   0   0   0   1   0   0   0   0   0
   0   0   0   0   0   0   0   1   0   0   0   0
   0   0   0   0   0   0   0   0   1   0   0   0
   0   0   0   0   0   0   0   0   0   1   0   0
   0   0   0   0   0   0   0   0   0   0   1   0
   0   0   0   0   0   0   0   0   0   0   0   1

octave:107> det(y)
ans = 1
octave:108> az = 2:7:98;
octave:109> az
az =

    2    9   16   23   30   37   44   51   58   65   72   79   86   93

octave:110> k = [2, 3, 7; 8, 3, 4];
octave:111> m = rot90(k, 3)  % Rotate 540° = 1 full rotation + 180°
m =

   8   2
   3   3
   4   7

octave:112> w = [1, 0, 1; 2, 3, 5];
octave:113> reshape(w, [], 1)
ans =

   1
   2
   0
   3
   1
   5
```




```
>> matAd
   17   18
   29   30
>>
```