SCS2211 - Laboratory II
Octave Lab Practical Sheet - 02

B.S.D. Silva

22001883

1)

```
Command Window
>> sqrt(98)
ans = 9.8995
>> 98^(1/2)
ans = 9.8995
>> 98^0.5
ans = 9.8995
>> nthroot(98,2)
ans = 9.8995
```

2)

```
Command Window
>> y = 7
y = 7
>> isinteger(sqrt(y))
ans = 0
>>
```

3)

```
Command Window
>> iskeyword("when")
ans = 0
>> iskeyword("while")
ans = 1
>> iskeyword("how")
ans = 0
>> iskeyword("which")
ans = 0
>> iskeyword("catch")
ans = 1
>> iskeyword("try")
ans = 1
>> iskeyword("until")
ans = 1
>> iskeyword("spmd")
ans = 1
>> iskeyword("spot")
ans = 0
>> iskeyword("partfor")
ans = 0
>> iskeyword("for")
ans = 1
>> iskeyword("global")
ans = 1
>> iskeyword("else")
ans = 1
>> iskeyword("e")
ans = 0
>> iskeyword("pi")
ans = 0
>> iskeyword("_FINE_")
ans = 0
>> iskeyword("__FINE__")
ans = 0
>> iskeyword("__LINE__")
ans = 1
>> iskeyword("break")
ans = 1
>> iskeyword("broke")
ans = 0
>> iskeyword("function")
ans = 1
>>
```

4)

```
Command Window                                                    ⊡ ✕
>> x=1
x = 1
>> y=2
y = 2
>> z=3
z = 3
>> who
Variables visible from the current scope:

x  y  z

>> whos
Variables visible from the current scope:

variables in scope: top scope

  Attr   Name         Size                     Bytes  Class
  ====   ====         ====                     =====  =====
         x            1x1                          8  double
         y            1x1                          8  double
         z            1x1                          8  double

Total is 3 elements using 24 bytes

>> |
```

5)

```
Command Window                                                    ⊡ ✕
>> clear
>> a = 5
a = 5
>> b = [1,2,3;4,5,6]
b =

   1   2   3
   4   5   6

>> whos
Variables visible from the current scope:

variables in scope: top scope

  Attr   Name         Size                     Bytes  Class
  ====   ====         ====                     =====  =====
         a            1x1                          8  double
         b            2x3                         48  double

Total is 7 elements using 56 bytes

>> |
```

6)

```
Command Window
>> clear
>> Inf / 2
ans = Inf
>> Inf / 0
ans = Inf
>> Inf / -2.3
ans = -Inf
>> Inf / -4
ans = -Inf
>> Inf / (1 + i)
ans =  Inf - Infi
>> |
```

7)

```
Command Window
>> Inf / Inf
ans = NaN
>> |
```

8)

```
Command Window
>> Inf^2
ans = Inf
>> |
```

9)

```
Command Window
>> sqrt(Inf)
ans = Inf
>> |
```

10)

```
Command Window                                              ⊡  ✕
>> Inf + Inf
ans = Inf
>> |
```

11)

```
Command Window                                              ⊡  ✕
>> Inf * Inf
ans = Inf
>>
```

12)

```
Command Window                                              ⊡  ✕
>> Inf - Inf
ans = NaN
>> |
```

13)

```
Command Window                                              ⊡  ✕
>> sqrt(-Inf)
ans =     0 + Infi
>> |
```

14)

```
Command Window
>> Inf ^ Inf
ans = Inf
>>
```

15)

```
Command Window
>> Inf / i
ans =  NaN - Infi
>>
```

16) clc Command will help to clear the screen from within Octave programs

17) By using clear command we can clear the terminal

18)

```
Command Window
>> clear
>> format short
>> pi + e
ans = 5.8599
>> format long
>> pi + e
ans = 5.859874482048838
>> whos
Variables visible from the current scope:
```

19)

```
Command Window
>> X = [2,3;4,1]
X =

   2   3
   4   1

>> A = [3,4,10;70,1,30]
A =

    3    4   10
   70    1   30

>> Z = [2,3;50,49;0,1]
Z =

    2    3
   50   49
    0    1

>> Y = [1,0,0;0,1,0;0,0,1]
Y =

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

20)

```
Command Window                                                    ⤢ ✕
>> X'
ans =

    2    4
    3    1

>> A'
ans =

    3   70
    4    1
   10   30

>> Z'
ans =

    2   50    0
    3   49    1

>> Y'
ans =

    1    0    0
    0    1    0
    0    0    1

>> |
```

21)

```
Command Window                                                    ⤢ ✕
>> fliplr(A)
ans =

   10    4    3
   30    1   70

>> fliplr(X)
ans =

    3    2
    1    4

>> fliplr(Y)
ans =

    0    0    1
    0    1    0
    1    0    0

>> fliplr(Z)
ans =

    3    2
   49   50
    1    0

>> |
```

22)

```
Command Window
>> flipud(A)
ans =

   70    1    30
    3    4    10

>> flipud(X)
ans =

   4    1
   2    3

>> flipud(Y)
ans =

   0    0    1
   0    1    0
   1    0    0

>> flipud(Z)
ans =

    0    1
   50   49
    2    3

>> |
```

23)

```
Command Window
>> x=7
x = 7
>> y=x^2 - 6*x + 5
y = 12
>> |
```

24)

```
Command Window
>> linspace(5,150,11)
ans =

 Columns 1 through 3:

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

 Columns 4 through 6:

   4.850000000000000e+01   6.300000000000000e+01   7.750000000000000e+01

 Columns 7 through 9:

   9.200000000000000e+01   1.065000000000000e+02   1.210000000000000e+02

 Columns 10 and 11:

   1.355000000000000e+02   1.500000000000000e+02
```

25)

```
Command Window
>> logspace(log10(10),log10(1000),7)
ans =

 Columns 1 and 2:

   1.000000000000000e+01   2.154434690031883e+01

 Columns 3 and 4:

   4.641588833612777e+01   1.000000000000000e+02

 Columns 5 and 6:

   2.154434690031885e+02   4.641588833612777e+02

 Column 7:

   1.000000000000000e+03

>> |
```

26)

```
Command Window
>> rand(3,4)
ans =

   0.948296334552371   0.958501930348098   0.809018465517576   0.637424609019094
   0.599016285584458   0.703544732798877   0.703131101321132   0.356247279326502
   0.703351526923345   0.116401907697372   0.854695578208103   0.593634523470091

>> rand(3,4)
ans =

 Columns 1 through 3:

   7.863470852062745e-01   8.805567063078895e-01   4.691070535441153e-01
   9.219796253225914e-01   6.253372020680652e-01   2.830485688459106e-02
   6.422051560122718e-01   9.252007592048859e-01   6.047030254264367e-01

 Column 4:

   2.107007358805129e-01
   5.626242259762555e-01
   4.338415483776467e-01

>> rand(3,4)
ans =

 Columns 1 through 3:

   7.707200031834673e-01   6.203684216382502e-01   7.912551476773836e-01
   7.977451527267332e-01   4.205396079882173e-02   1.983419584458184e-01
   7.392177094452355e-01   8.069779251525500e-02   6.234338044388378e-01

 Column 4:

   2.077826594061359e-01
   3.724454613200215e-01
   6.834815361253200e-01

>> |
```

27)

```
Command Window
>> x=rand(3,4)
x =

 Columns 1 through 3:

   8.171912806032893e-01   3.567559200216690e-01   5.554566858405180e-01
   1.957372359807054e-01   6.703477404351622e-01   3.075136679315077e-01
   5.738656558769983e-02   6.483120611040349e-02   6.114793451665963e-01

 Column 4:

   1.535453411351850e-01
   5.534453625634626e-01
   1.288297213870790e-01

>> x=rand(3,4)
x =

   0.529355434431819   0.398231311192574   0.478303770121917   0.871272849344765
   0.859225737773288   0.315154699082536   0.239408277363647   0.256105922749050
   0.449778186280548   0.168557736556714   0.596307700148010   0.559115536601398

>> x=rand(3,4)
x =

 Columns 1 through 3:

   6.271029338693525e-01   7.996787204287180e-01   8.637689634324172e-01
   4.931025250311636e-01   1.364712593073009e-01   2.082617746855024e-01
   9.776429079472366e-01   6.923443695784459e-01   8.144225214070233e-02

 Column 4:

   2.713515989387905e-02
   1.254620456310411e-01
   4.701927652067015e-01

>> |
```

28)

```
Command Window
>> x = 0
x = 0
>> x = x + 25
x = 25
>>
>> x = x + 25
x = 50
>> x = x + 25
x = 75
>> x = x + 25
x = 100
>> x = x + 25
x = 125
>> x = x + 25
x = 150
>>
>> x = x + 25
x = 175
>> x = x + 25
x = 200
>> x = x + 25
x = 225
>> |
```

29)

```
Command Window
>> a = 2
a = 2
>> z = 8
z = 8
>> a = 2*z + 4
a = 20
>> a = 2*z + 4
a = 20
>> a = 2*z + 4
a = 20
>> a = 2*z + 4
a = 20
>> a = 2*z + 4
a = 20
>> a = 2*z + 4
a = 20
>> a = 2*z + 4
a = 20
>> a = 2*z + 4
a = 20
>> a = 2*z + 4
a = 20
>> a = 2*z + 4
a = 20
>> a = 2*z + 4
a = 20
>> a = 2*z + 4
a = 20
>> a = 2*z + 4
a = 20
>> a = 2*z + 4
a = 20
>> z
z = 8
>> a
a = 20
>>
```

30)

```
Command Window
>> A = [2,3;4,5]
A =

   2    3
   4    5

>> inv(A)
ans =

  -2.500000000000000   1.500000000000000
   2.000000000000000  -1.000000000000000

>>
```

31)

```
Command Window                                                    ⊡ ✕
>> az = 2:7:98;
>> az
az =

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

>> az
az =

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

>> |
```

32)

```
Command Window                                                    ⊡ ✕
>> w = [1,0,1;2,3,5]
w =

   1   0   1
   2   3   5

>> x = w(:);
>> x
x =

   1
   2
   0
   3
   1
   5

>> |
```

33)

```
Editor                                                            ⊡ ✕
File   Edit   View   Debug   Run   Help

matAd.m ✕
1   H = [2,3;4,5]
2   K = [1,0;5,6]
3   V = H * K
4   V - [1,2;2,2]
```

34)

```
Command Window                                                          ⧉ ✕
>> matAd                                                                 ▲

H =

    2    3
    4    5

K =

    1    0
    5    6

V =

   17   18
   29   30

ans =

   16   16
   27   28

>> |
```

35)

```
Command Window                                                          ⧉ ✕
>> w = [1, 0, 1; 2, 3, 5];                                               ▲
>> vertical_vector = w(:);
>> disp(vertical_vector);
   1
   2
   0
   3
   1
   5
>> |
```

36)

```
Command Window                                                          ⧉ ✕
>> matAd;                                                                ▲
   17    18
   29    30
>> |
```