ONE DIMENSIONAL ARRAYS(N DIMENSIONAL VECTORS)

%We have seen that MATLAB row vectors are equivalent to a 1 dimensional array. We will look at this topic in more detail.
An array is a variable with many components all with the same name but We can reference with a **subscript variable or index** representing the **position in the array.** For example before when we set up the 't' row vector as in

DEMO IN CLASS

```
» t=0:.1:1
t =  Columns 1 through 7
        0    0.1000    0.2000    0.3000    0.4000    0.5000    0.6000
    Columns 8 through 11
    0.7000    0.8000    0.9000    1.0000
```

% We just set up an eleven component array. (1 row of 11 columns in Matrix notation). We can think of the variable t as having 11 single values which can be referenced by their position in the vector (or array). We reference each member uniquely by specifying the column number within the array name. The column number is said to **index** the position. Some authors call the **index** a **subscript** reference to the array.
In the case above note how we do this with the following examples.

```
» t(1)  = 0
» t(2)  = 0.1000
» t(3)  = 0.2000
» t(10) = 0.9000
» t(11) = 1
>> t(12)   Index exceeds matrix dimensions.
Size of the matrix is said to be 1 X 11
>>size(t)   ans= 1  11
```

% Here the numbers in parenthesis basically indicate the position in the array. We can use a general variable to reference position as variable 'i' in this next example. The variable i is said to be an **index** or **subscript** variable.
NOTE: In the examples to follow we will often use one line forms for statements. These are useful as you gain proficiency in the computer language. Basically comma's are used to separate the parts of statements we were previously using on another line with indentation. Study the examples carefully.

DEMOS

```
» for i=1:10,fprintf(' %6.4f ',t(i)),end
0.0000 0.1000 0.2000 0.3000 0.4000 0.5000 0.6000 0.7000 0.8000 0.9000

% An example that will initializing the first five members of the array 'x'.
» n=5
» for i=1:n,x(i)=3+i,end;
»x =    4     5     6     7     8
% we have created a five component array

% Initializing another array
» for i=1:5,j(i)=10,end;
»j =    10    10    10    10    10

% and another initialization
» for k=1:4,dist(k)=k*1.5,end;
»dist =   1.5000    3.0000    4.5000    6.0000
```

```
% read values from keyboard for an array
i=1
while i< 4
   n(i)=input('enter array value');
   i=i+1;
end;


 enter array value 45.67
 enter array value 34.12
 enter array value 888.9

»n =   45.6700   34.1200  888.9000
```

% We can use output or input functions to reference individual members(or components) or all members of the array as these illustrations of the disp() function show.

```
» disp(n)
   45.6700   34.1200  888.9000

» disp(n(1))
   45.6700

» disp(n(2))
   34.1200
```

% A short program segment to sum the elements of above array is now illustrated, which uses the echo to show the growth of the sum and final average.

```
 sum=0;
 for i=1:3,sum=sum+n(i),end;
 avg =sum/3
```

% A mathematical example of using one dimensional arrays in m-file called

DEMO M FILE

```
fibno.m
» type fibno
% An M-file to calculate Fibonnaci numbers
% We first load the starting two elements with 1's; Note the use of the
brackets which equivalently load f(1) and f(2) with a 1!
f=[1 1];
i=1;
while f(i)+f(i+1)<1000
  f(i+2)=f(i)+f(i+1);
  i=i+1;
end
% output
for j=1:i+1,fprintf('%3.0f ',f(j)),end;
% Running the program
» fibno
  1   1   2   3   5   8  13  21  34  55  89 144 233 377 610 987 »
```

f is a row vector of 16 columns!

% NOTES Since the fprintf() format part did not have the special character '\n', all values which were output one at a time in the loop appeared on the same line. The final value would cause MATLAB to begin when the program ends just after

A few word on Symbolic (algebraic) computations

If you have the symbolic Math Toolbox in the version.

DEMO SYMBOLIC TOOLBOX
>>Help symbolic will work and give you lots of info

Short illustration of this toolbox

syms x y

>> (x-y)^2 *(x-y)^3

ans =(x - y)^5

>> expand(ans)

ans =x^5 - 5*x^4*y + 10*x^3*y^2 - 10*x^2*y^3 + 5*x*y^4 - y^5

>> factor(ans)

ans = (x - y)^5

Simplify(ans) would have done the same for the last example.
But consider the following which is not easily factored.

>> (x^3-y^3)/(x-y)

ans =(x^3 - y^3)/(x - y)

>> expand(ans)

ans =x^3/(x - y) - y^3/(x - y)

>> simplify(ans)

ans =x^2 + x*y + y^2

LABORATORY TASKS (print m files and outputs and any data files created. Number all tasks).Remember use comments to you and the user.

28(4 PTS) Set up a program that puts the square of numbers from 1 to 20 in **a one dimensional arra**y. The program will also sum all members of the array as well as compute the product of all members. Output the values of the sum and product of all members. *Do not use intrinsic functions sum() and prod(.), this is not an array calculation. The lesson is about using the index of an array so set up a **"for"** loop to do the calculations, which means each value is obtained one at a time, as well as the sum and the product but still creates an array..* .

29(3 PTS) The MATLAB symbolic toolbox is used when you do the following.
Define x,y as symbolic variables
Find the factors of the equation in the quadratic equation that follows
$6x^2 - x - 35=0$ and from the answer state what the solution(s) would be

Find factors of $8x^5 + 4x^4 y - 10x^3y^2 - x^2y^3 + 4xy^4 - y^5$
Assume the latter is equal to zero what is the relation(s) of the variable x to y that would satisfy that equation.
*NOTE: in both cases =0 **do not** use Matlab intrinsic functions, obtain the answer by hand and write into the submittal. Print all up to show you did the work.*