5- for loops

The for loop repeats a group of statements a fixed, predetermined number of times. A matching end delineates the statements. The general form of a for statement is:

```
for variable = x : s : y,
    statement, ..., statement
end
```

The variable and then the following statements, up to the end, are executed. The expression of the form x : s : y are being to be a vector represents the beginning, x, and the end, y, of the loop by a step of, s. Note that you may not want s if the step size was 1.

Some examples will be listed here to get a brief knowledge a bout for loops:

Example: Suppose we want to find the factorial of n (n!):

```
n = input ('Enter any number:');
f=1;
for i = 1:n
   f = f * i ;
end
disp ('The factorial of this number is:')
f
```

Example: Find 100 values of x and y obtained from the difference equations:

$$\mathbf{x_{k+1} = y_k (1 + \sin(0.7x_k)) - 1.2\sqrt{|x_k|}}$$

$$\mathbf{y_{k+1} = 0.21 - x_k}$$

starting with $x_o=y_o=0$.

```
x(1)=0;y(1)=0;
for k=1:10
   x(k+1)=y(k)*(1+sin(0.7*x(k)))-1.2*sqrt(abs(x(k)));
   y(k+1)=0.21-x(k);
end
```

Note: MATLAB did not accept zero indices like x(0) or y(0) as an example.

6- while loop

The while loop repeats a group of statements an indefinite number of times under control of a logical condition. A matching end delineates the statements. The general form of a while statement is:

```
while expression
   statements
end
```

So the statements are executed while the real part of the expression has all non-zero elements. The expression is usually the result of a logical expressions (==, <, >, <=, >=, or ~=).

Example: Write a script file to find a solution for the polynomial ($x^3 + x - 3=0$) by using Newton's method. Give an initial guess to x and stop the program either when the absolute value of y(x) is less than 10^{-8}, or after 20 steps.

Hint: Newton's method used to solve a general equation y(x)=0 by repeating the assignment:

$$\mathbf{x_{k+1} = x_k - \frac{y(x_k)}{y'(x_k)}}$$

where $\mathbf{y'(x_k)}$ (i.e. $\mathbf{\frac{dy}{dx}}$) is the first derivative of $\mathbf{y(x_k)}$. The process continues until $\mathbf{y(x_k)}$ is close enough to zero.

The solution of this problem will be as follow:

```
% Newtons Method
steps=0;
x=input('initial guess:')
y=x^3+x-3;
e=1e-8;
while(abs(y)>=e)&(steps<20)
```

```
    y=x^3+x-3;
    y_dash=3*x^2+1;
    x=x-(y/y_dash);
    steps=steps+1;
    disp([x y])
end
```

Note that there are two conditions that will stop the while loop: convergence, or the completion of 20 steps. Otherwise the script could run indefinitely.
Here is a sample run (with format long), starting with initial guess of x = 1.

```
x =
     1
   1.25000000000000  -1.00000000000000
   1.21428571428571   0.20312500000000
   1.21341217578282   0.00473760932945
   1.21341166276241   0.00000277908667
   1.21341166276223   0.00000000000096
```

7- <u>continue statement</u>

The continue statement passes control to the next iteration of the for or while loop in which it appears, skipping any remaining statements in the body of the loop. In nested loops, continue passes control to the next iteration of the for or while loop enclosing it.

Example: Write a script file to print the even elements in matrix A. Where:

$$\mathbf{A} = \begin{bmatrix} \mathbf{23} & \mathbf{11} & \mathbf{12} & \mathbf{34} \\ \mathbf{42} & \mathbf{56} & \mathbf{2} & \mathbf{9} \\ \mathbf{77} & \mathbf{82} & \mathbf{52} & \mathbf{21} \\ \mathbf{12} & \mathbf{10} & \mathbf{33} & \mathbf{2} \end{bmatrix}$$

```
a=[23 11 12 34;42 56 2 9;77 82 52 21;12 10 33 2];
for i=1:4
   for j=1:4
      if rem(a(i,j),2)~=0
         continue
     end
     disp(a(i,j))
   end
end
```

8- break statement

The break statement lets you exit early from a for or while loop. In nested loops, break exits from the innermost loop only.

Example: Write a script file to find a solution for the exponential series below.

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots\ldots + \frac{x^n}{n!}$$

Make the output precision be: 0.0001

Hint: The program will stopped when the value of ($\frac{x^n}{n!}$) reached to 0.0001 even when the counter not reached
its final value.

```
% This script is used to find the solution of the exponential series exp(x)
x=input('Enter the value of x:')
n=input('Enter the highest exponent (n):')
s=0;
for i=1:n
   f=1;
   for j=1:i;
      f=f*j;
```

```
    end
    f1=f;
    e=x^i/f1;
    if e<0.0001
       break
    end
    s=s+e;
end
disp('the result is:')
s
```

Exercises:

1. If C and F are the Celsius and Fahrenheit temperature respectively, the formula for conversion from Celsius to Fahrenheit is:

$$\mathbf{F = (9C / 5) + 32}$$

Write a script which will ask you for the Celsius temperature and display the equivalent Fahrenheit one with the following comments:

“Cold” when $F \leq 41$.
“Nice” when $41 < F \leq 77$.
“Hot” when $F > 77$.

2. Set up any 4×4 matrices A & B. Write some statements to execute the following statements:

a) A+B if A = B.
b) A^2+B^2 if |A| > |B|.
c) $\sqrt{A} + (\sqrt{B})^3$ if all the eigen values of A are nonzero.

3. Write a program to compute the below functions depending on your entry from 1 to 3:

1. **x(t) = sin(t)+tan(t)**
2. **x(t) = cosh(t)**
3. **x(t) = $\tan^{-1}$(4t)**

Use the interval $-2\pi \le t \le 2\pi$ in steps of $\pi/2$.

4. Write a script file to find **y** with respect to all variables:

a) $y = \dfrac{n!}{(n-r)!}$

b) $y = \sum_{k=1}^{100} \dfrac{3x_k}{(2x_k + x_k^2)}$

5. When a resistor (R), capacitor (C) and battery (V) are connected in series, a charge Q builds up on the capacitor according to the formula:

$$Q(t) = CV\,(1-e^{-t/RC})$$

If there is no charge on the capacitor at time t=0. The problem is to monitor the charge on the capacitor every 0.1 seconds in order to detect when it reaches a level of 8 units, given that V=9, R=4 and C=1. Write a program which displays the time and charge every 0.1 seconds until the charge first exceeds 8 units (i.e. the last charge displayed must exceed 8).

6. A square wave of period T may be defined by the function

$$f(t) = \begin{cases} 1 & (0 < t < T) \\ -1 & (-T < t < 0) \end{cases}$$

The Fourier series for f(t) is given by:

$$F(t) = \frac{4}{\pi} \sum_{k=0}^{\infty} \frac{1}{2k+1} \sin\left[\frac{(2k+1)\pi t}{T}\right]$$

It is of interest to know how many terms are needed for a good approximation to this infinity sum. Taking T=1, write a program to compute and display the sum to n terms of the series for t from 0 to 1 in steps of 0.1, say. Run the program for different values of n, e.g. 1, 3, 6, etc.

7. Write a program to compute a table of the function

$$f(x) = x\sin\left[\frac{\pi(1+20x)}{2}\right]$$

over the closed interval [-1,1] using increments in x of (a) 0.2 (b) 0.1 and (c) 0.01.

8. One of the fastest series for ($\pi/4$) is:

$$\frac{\pi}{4} = 6\tan^{-1}\left[\frac{1}{8}\right] + 2\tan^{-1}\left[\frac{1}{57}\right] + \tan^{-1}\left[\frac{1}{239}\right]$$

Use the series below to compute $\tan^{-1}(x)$:

$$\tan^{-1}(x) = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \frac{x^9}{9} - \ldots\ldots + \frac{x^{99}}{99}$$

9. Find 14 values of **S** and **R** obtained from the difference equations:

$$S_{k+1} = R_k(1/\cos(0.3S_k))$$
$$R_{k+1} = 0.4S_k + S_k^2$$

starting with $S_o=R_o=1$.

10. Write a script file to find a solution for the polynomial ($x^4+2x^2+4x-5=0$) by using Newton's method. Give an initial guess to x and stop the program either when the absolute value of f(x) is less than 10^{-5}, or after 100 steps.

11. Write a script file to print the odd elements in matrix B. Where:

$$B = \begin{bmatrix} 3 & 1 & 32 & 54 \\ 9 & 44 & 20 & 98 \\ 72 & 12 & 55 & 31 \\ 87 & 90 & 23 & 2 \end{bmatrix}$$

12. Write a script file to find a solution for the exponential series below.

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \ldots\ldots + (-1)^{n+1} \cdot \frac{x^{2n-1}}{(2n-1)!}$$

Input x and make the output precision be: 10^{-6}