E77 Midterm Examination 2

November 3rd, 2006

NAME : [redacted]

SID : [redacted]

SECTION : **1** or **2** (please circle your lecture section)

LAB :

#11: TuTh 8-10	#12: TuTh 10-12	#13: TuTh 12-2	#14: TuTh 2-4
#15: TuTh 4-6	#16: MW 8-10	#17: MW 10-12	#18: MW 2-4
#19: MW 4-6			

(please circle the lab section in which you are enrolled)

Part	Points	Grade
A	10	9
B	8	8
C	6	5
D	8	4.5
E	7	7
F	6	6
TOTAL	**45**	39.5

Notes:
1. Write your name on the top right corner of each page.
2. Record your answers only in the spaces provided.
3. You may not ask questions during the exam.
4. You may not leave the exam room before the exam ends.

 Name: [redacted]

3

Part A (*10 points*)

A.1 (*3 points*) Complete the following MATLAB function, called `currency_conv`.

```
function Aout=currency_conv(Ain,Tin,Tout)
% The function converts money from one currency to
% another.
%   Ain : input amount
%   Tin : input currency type
%   Tout: currency type for output
%   Aout: output amount (to be determined)
% Types of currency are denoted by a letter as follows:
%   D for Dollars
%   E for Euros
% The exchange rate is: 1 dollar (D) = 0.75 Euros (E)

if Tin==Tout

  Aout = Ain;

else

  switch Tin

% Dollars to Euros

    case 'D'                          % add code here

    Aout= .75*Ain;                    % add code here

% Euros to Dollars

    case 'E'                          % add code here

    Aout= 4/3*Ain;                    % add code here

  end

end
```

2

 Name: [redacted]

A.2 (*3 points*) Given the function

$$y(x) = -3x^5 + 3x^3 ,$$

complete the following MATLAB script to plot y (vertical axis) as a function of x (horizontal axis), for x ranging between -1 to 1, with an increment of 0.1. Plot a blue star at points where the derivative of $y(x)$ is greater than or equal to zero, and plot a red star at points where the derivative of $y(x)$ is less than zero.

```
x = -1:0.1:1;

y = -3*x.^5 + 3*x.^3;

plot(x,y)

hold on

% yp is the derivative of y

yp = polyder(y);                    -1   % add code here

for k= 1:length(x)                  ✓    % add code here

  if polyval(yp,x(k)) >= 0          ✓    % add code here

    plot(x(k),y(k),'b*')

  else

    plot(x(k),y(k),'r*')

  end

end
```

4

Name: [redacted]

A.3 (*4 points*) Let

```
>> A = [1 2 3 4];
>> B = [1 0];
>> X = 1:4;
```

Record the output of each of the following MATLAB commands:

```
>> C = conv(A,B)
```

C = [1 2 3 4 0] ✓

```
>> D = polyval(B,X)
```

D = [1 2 3 4] ✓

```
>> E = A + 2*[B B]
```

E = [3 2 5 4] ✓

 Name: [redacted]

Part B (*8 points*)

The arrays A and B are generated by the MATLAB code below:

```
A = zeros(3,3);

B = zeros(3,1);

for k= [3 1 2]

   for m=0:2

      A(k,m+1) = A(k,m+1) + k + m;

      for n=1:2

         B(k) = B(k) + n;

      end

   end

end

A

B
```

B.1 (*4 points*) What are the values of A and B after execution of the above code?

A =

$$\begin{bmatrix} [illegible] & [illegible] & [illegible] \\ [illegible] & [illegible] & [illegible] \\ 3 & 4 & 5 \end{bmatrix}$$

B =

$$\begin{bmatrix} [illegible] \\ 9 \\ 9 \end{bmatrix}$$

4

 Name: [redacted]

B.2 (*4 points*) Consider the following system of linear algebraic equations:

$$
\begin{aligned}
x_1 + 2x_2 + x_3 &= 5 \\
x_1 &= 1 \\
x_1 + 3x_3 &= 13
\end{aligned}
$$

The system can be put in the form $[A][x] = [b]$, where $[A]$ is a 3×3 array, $[b]$ is a 3×1 array and

$$
[x] = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}.
$$

Input $[A]$ and $[b]$ in MATLAB format below:

```
>> A = [1 2 3; 1 0 0; 1 0 3]      % add code here

>> b = [5; 1; 13]                 % add code here
```

Also, identify two different commands for solving this system in MATLAB:

```
>> x = A\B                        % add code here

>> x = inv(A)*B                   % add code here
```

Part C (*6 points*)

Consider a particle on the x-axis with position $x = 0$ at time $t = 0$. After each unit of time, there is a 50% probability that the particle moves one unit to the left and a 50% probability that the particle moves one unit to the right. Thus, at time $t = 1$, the possible positions of the particle are $x = -1, +1$, and at time $t = 2$, the possible positions of the particle are $x = -2, -1, 0, +1, +2$, etc. A sequence of particle positions for times $t = 0, 1, 2, \ldots, s$ is called a *trajectory* of length s.

Complete the MATLAB function below so that it outputs n possible particle trajectories, each of length s. (The trajectories are stored as columns of a single $s \times n$ array T). The function should also output the number R of these trajectories for which the particle is ever to the right of the input value L.

```
function [T,R]=particle(s,n,L)

T = zeros(n,s);   s, n

for T_row = 2:n   2:s

    for T_col = 1:n                                    % add code here

        right_or_left = rand;

          if (right_or_left > 0.5)

               T(T_row,T_col) = T(T_row-1,T_col)+1      % add code here

          else

               T(T_row,T_col) = T(T_row-1,T_col)-1      % add code here

          end

    end

end

maxT = max(T);

R = sum(find(maxT > L))                                 % add code here
```

 Name: [redacted]

Part D (*8 points*)

A sequence of polynomials $P_n(x)$ can be defined by the recursive formula

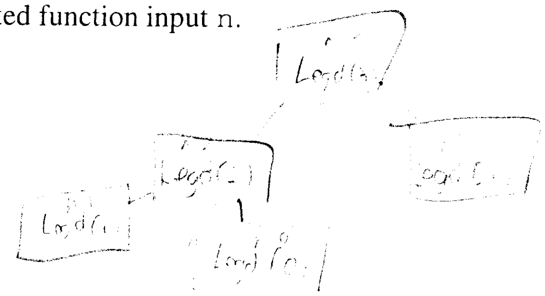
$$nP_n(x) = (2n-1)xP_{n-1}(x) - (n-1)P_{n-2}(x) \quad , \quad n = 2, 3, \dots ,$$

with $P_0(x) = 1$ and $P_1(x) = x$.

D.1 (*4 points*) Complete the MATLAB function Legd below that determines P_n from the equation given above for any non-negative integer input.

```
function y = Legd(n)
% The function uses recursion to return the
% coefficients of the n-th degree polynomial.
if n==0
    y = 1 ;                                     % add code here
elseif n == 1                                   % add code here
    y = X ;                                     % add code here
else
  y=(2*n-1)*x*Legd(n-1)-(n-1)*Legd(n-2)         % add code here
end
```

D.2 (*1 point*) Draw the recursive tree for Legd(3). Recall that each node of the tree corresponds to a call of the function Legd. In each node, write the value of the associated function input n.



D.3 (*1 point*) List the values of n in the pre-order traversal of the tree in Part D.2.

Answer:

D.4 (*1 point*) List the values of n in the post-order traversal of the tree in Part D.2.

Answer: 0 2, 1, 3

D.5 (*1 point*) What is the polynomial $P_2(x)$?

Answer: $3x^2 - 1$

7

Part E (*7 points*)

Consider the following MATLAB function:

```
function y=printrec(n)
% A mystery function that employs recursion
% rem(a,b): remainder after division of a by b
% floor(a): round to nearest integer toward minus infinity

if n<0

  n = -n;

end

if (floor(n/10) ~= 0)

  y = [printrec(floor(n/10)) rem(n,10)];

else

  y = [n];

end
```

E.1 (*5 points*) Write the output of the following MATLAB calls:

```
>> printrec(129)

ans = [1 2 9]

>> printrec(-129)

ans = [1 2 9]
```

E.2 (*2 points*) In the preceding function, replace the line

```
y = [printrec(floor(n/10)) rem(n,10)];
```

by

```
y = [rem(n,10) printrec(floor(n/10))];
```

Assuming that this change has been implemented, write the output of the MATLAB call:

```
>> printrec(129)

ans =   [9  2  1]
```

 Name: [redacted]

6

Part F *(6 points)*

Consider the MATLAB function f defined below:

```
function flag = f(a)

flag = 0;     % Initialize flag

k = 1;        % Initialize counter

while ((k <= length(a)) & (flag == 0))

  j = 1;

    while ((j <= k-1) & (flag == 0))

      if (a(k) == a(j))

        flag = flag + 1;

      end

      j = j + 1;

    end

    k = k + 1;

 end
```

Determine the output of this function in the following two cases:

```
>> f([1 4 5 7 8 9 10])

ans = 0

>> f([7 6 4 3 2 4 2 9 0])

ans = 1
```