Name:____________________________________

ID#:______________________________________

Enrolled Discussion (circle one): M–10am M–11am M–7pm F–10am F–11am F–12pm

MAE10
Midterm Examination II
Winter Quarter 2010

Instructions: You have 90 minutes to complete the exam. Notes on **two** sides of an 8.5''x11'' sheet of paper are allowed. Closed book. No calculators or electronic devices of any kind.

Section 1: Short answer. (2 points each)

(1.1) If your function is called mycoolfunction, what must be the name of the m-file that contains it?

(1.2) Why was the UNIVAC computer important?

(1.3) How many arguments can you pass down to a function?

(1.4) Within a particular program, how many times can you call a function?

(1.5) What is the difference between the %f, %g, and %e formatting commands?

Section 2: Identify and briefly explain any and all ***errors*** that would prevent the code from executing in the following MATLAB programs. Warnings and “bad programming” are not considered errors – those would not prevent the code from running. If you believe there are no errors, write “OKAY”. (3 points each)

(2.1)

```
x = [2, 2, 4, 5];
y = [33, 98]
z = [11, 11, 11];
if( x(3) > y(1) | z(3) ~= z(3) )
   y(1) = z(3);
elseif( x(4) < y(3)  &  x( 4 ) == z(2) )
   z(1) = x(3);
elseif( z(1) == y(1)  |  x(3) == y(1) )
   fprintf('%7.2f\n', z)
end
```

(2.2)

```
x = -1;
y = 5;
for i=1:1:x
   y = y+1;
   disp(x+y)
end
```

(2.3)

```
z = 100;
while(z < 10)
   z = z - 1;
   if(z == z | z > z)
         z = z;
   end
end
```

(2.4)

```
sport = 'baseball' ;
switch sport
   case { football, sport }
         disp( sport )
   otherwise
         disp( 'hello' )
end
```

(2.5)

```
x = 4;
y = 5;
fprintf('The value of %7.2f is %7.2f', x, x)
fprintf('y is %10.2e', x)
```

(2.6)

```
x = [3.2 , 6.1 , 4 , pi];
fprintf( '%g   \n %e', x(1:2) , x(4) )
fprintf( '%i    %10.1g    %f', x )
```

For problems (2.7) through (2.10), I provide a function and the “main” program that calls the function, each stored in a different M-File. Assume both M-Files are in the same directory. Errors may exist in either or both programs.

(2.7)

In the “main” program:

```
x = 5;
y = 4;
fprintf('%7.2f', zot(x,y), zot(y,x) )
```

In the m-file containing the function `zot`:

```
function [ result ] = zot(a,b)
a = 100;
result = a + b;
return
end
```

(2.8)
In the “main” program:

```
x = [1, 3, 5, 7]
h = [100, 200 ]
y = apple(x) / apple(h)
```

In the m-file containing the function `apple`:

```
function [ result ] = apple(core)
result = 2;
for i=1:numel(core)
  result = result + core(i) + 1;
end
end
```

(2.9)
In the “main” program:

```
x = [3, 4, 5, 5];
y = [6, 7];
[a, b] = cheese(x, y);
```

In the m-file containing the function `cheese`:

```
function [ results ] = cheese(y, x)
results = 0;
return
for i=1:numel(y)
  results = x(i) + y(i);
end
end
```

(2.10)
In the “main” program:

```
x = [3, 4, 5, 6 ; 7, 8, 9, 10];
h = pig( x(3,:) ) * pig( x(:,3) )
```

In the m-file containing the function `pig`:

```
function [ x ] = pig(y)
x = -10;
x = x + mean(y);
results = 10;
end
```

Section 3: Write the ***exact*** output that will be produced by each of the following programs. Assume that there are no errors. Clearly distinguish answers from any scratch work and indicate which line of code each answer corresponds to. If you are given specific formatting instructions, you must **use underscores to indicate any and all blank spaces** (one underscore per blank space). In addition, carriage returns (newline) should be clear. (4 points each)

(3.1)

```
a = ( -1 : +2 : +5 );
b = ( +5 : -2 : -1 );
for i=1:3
   summy(i) = a(i) + b(i+1);
end
fprintf('%3i', summy)
```

(3.2)

```
c = [1, 5, 7, 8, -10];
d = [9, 5, 7];
i = 1;
while(i < d(2))
   m(i) = i + c(i);
   i = i + 1;
end
fprintf('%3i %5i\n', m )
```

(3.3)

```
x = (1.25: 3.25);
y = x + 2;
table1 = [x , y ];
table2 = [x ; y ];
fprintf('%7.2f %5.2f\n' , table1)
fprintf('%7.2f %5.2f\n' , table2)
```

(3.4)

```
a = [1, 8, -1, -9];
b = [-5, 8, 1, -6, 7, 9];
for i=1:numel(a)
   c(i) = 0;
   for k = i : numel(a)
         c(i) =  c(i) + a(k) + b(k);
   end
end
fprintf('%7.2f\n', c)
```

For problems (3.5) through (3.7), I provide a function program and the “main” program that calls the function. Assume both programs are in the same directory.

(3.5)

In the “main” program:

```
        a = 2.5;
        b = 9.5;
        y = 5.5;
        z = 6.5;
        [ a , b ] = funky(a, b);
        fprintf('a is %7.3f\n', a)
        fprintf('b is %7.1f\n', b)
```

In the m-file containing the function `funky`:

```
        function [ y , z ] = funky(b, a)
        z = b + a;
        a = b;
        b = a;
        y = a + b;
        fprintf('a is %6.3f\n', a)
        fprintf('b is %6.1f\n', b)
        end
```

(3.6)

In the “main” program:

```
        even = [1, 3, 5, 7];
        odd = [2, 4, 6, 8];
        [ a , b ] = chunky( even(1:3) , odd(1:2) );
        fprintf('%3i\n', a )
        fprintf('%4i\n', b )
```

In the m-file containing the function `chunky`:

```
        function [ x , y ] = chunky ( ev, od)
        for i=1:numel(od)
          x(i) = od(i) + ev(i);
          y(i) = od(i) - ev(i);
        end
        end
```

(3.7)

In the “main” program:

```
        a  = 4.5;
        b  = 5.5;
        c  = [1.1, 2.2 ];
        x  = f1(a, b) +  f2(b);
        y  = f1( c(1), b ) -  f2( c(2) ) ;
        fprintf('x = %7.2f\n', x  )
```

In the m-file containing the function `f1`:

```
        function [ output ] = f1 (x, y)
        output = x - f2(y);
        end
```

In the m-file containing the function `f2`:

```
        function [ output ] = f2 (x)
        fprintf('x = %7.2f\n', x  )
        output = x + 2;
        end
```

Section 4: Write a MATLAB program to solve each of the following problems. You do not have to write the output of the code.

(4.1) Create a program that calculates the value of the expressions x and y,

$$x = t + t^2$$
$$y = t^3 + 3$$

from t = 0 seconds to 1 second in increments of 0.1 seconds. Display the values of t, x, and y in a three-column table with special formatting. The variable t (in the first column) should be displayed using six total spaces, two of which are for digits to the right of the decimal point. The variable x (in the second column) should be displayed using seven total spaces, three of which are for digits to the right of the decimal point. The variable y (in the third column) should be displayed using six total spaces, three of which are for digits to the right of the decimal point. (4 points)

The first three lines of the table would look like the following:

```
  0.00  0.000 3.000
  0.10  0.110 3.001
  0.20  0.240 3.008
```

(4.2) A program calls a function called `analyzeT`:

```
[ maxT , minT , avgT ] = analyzeT( temp );
```

Write the function `analyzeT` that takes as input an array of temperatures called `temp`. The function returns as output the maximum temperature (`maxT`), the minimum temperature (`minT`), and the average temperature (`avgT`). For example, if `temp` contains `[90,91,95,92]`, `maxT` would be 95, `minT` would be 90, and `avgT` would be 92.

You cannot use MATLAB's built-in functions max( ), min( ), sum( ), or mean(). (5 points)

(4.3) A program calls a function called switcharoo:

```
d = switcharoo(c)
```

Write the function called switcharoo that takes as input an array c and returns as output an array d whose elements are in reverse order of the input array. For example, if array c contains `[1,3,7,7]` the array d would contain `[7,7,3,1]`.

You cannot use any of MATLAB's built-in functions for this problem except functions that determine the number of elements in an array. (5 points)

(4.4) A program in a vending machine uses a function called `change`:

```
[quarter, dime, nickel, penny] = change(money)
```

Write the function called `change` that takes as input an amount of cents (stored in the variable `money`) and calculates the number of quarters, dimes, nickels, and pennies to dispense to the user. These quantities are returned as output and stored in the variables `quarter, dime, nickel,` and `penny`. The total number of coins dispensed to the customer should be <u>minimized</u> (you cannot make the vending machine only dispense only pennies, unless money is between 1 and 4 cents). For example, if `money` is 132 cents, then `quarter` would be 5, `dime` would be 0, `nickel` would be 1, and `penny` would be 2. (6 points)