

Computer Aided Analysis for Electrical Engineering (Eng103)
City College of New York - Spring 2003

EXAM I

Instructor: Khalil

Print Name:
Family (Last)

.....
Given (First)

Part A: Each problem of this type is worth five points.

For each of the following Matlab statements (or sequence of Matlab statements) write out what is contained in the variable called z. If the statements cause an error, indicate that instead of writing any value(s) for z.

1. `>> z = 0:2:10`

2. `>>x = [1 3 5];`
 `>>z = 3 -4*x`

3. `>> x = 0: pi/2: 2*pi;`
 `>> z = sin(x)/cos(x)`

4. `>> x = linspace(0,1,41);`
 `>> z = x(1:3:39)`

5. `>>A =[1 2 3;4 5 6;7 8 9];`
 `>> z = A(:,2)`

Part B:

The following problems require you to demonstrate an understanding of Matlab m-file functions. You should provide either an explanation of how you obtained the answer, or some sample computations that demonstrate why your solution is correct.

6. [10 points]: The *sumpos* function listed below contains a bug

```
function s = sumpos(x)
% sumpos Add up the positive elements of a vector
% Synopsis: s = sumpos(x)
% Input: x = a row or column vector
% Output: s = sum of elements in x that are greater than zero
s = 0;
for i = 1:length(x)
    if x > 0
        s = s + x;
    end
end
end
```

The following Matlab session shows the result of running *sumpos* for three different inputs.

```
>> sumpos(3)
ans =
3
>> sumpos(-5)
ans =
0
>> sumpos(-5:2:5)
ans =
0
```

The correct answer for the *sumpos*(-5:2:5) is 9. Identify, and fix the error(s) in the *sumpos* function. Do not use Matlab's built-in *sum* function.

[5 points Extra Credit]: Write another version of the *sumpos* by eliminate the loop using a vectorized expression that returns the same result as the correct *sumpos* function. For the extra credit, using the *sum* function is allowed.

7. **[15 points]:** Consider the *myfun* function:

```
function z = myfun(n)
y = 0;
for i=1:n
    x = 2*i;
    if (i/2-fix(i/2)) > 0.1
        y = y + sqrt(x);
    else
        y = y + x;
    end
end
z = y/x;
```

What is the value of *s* returned by the expression

```
>> s = myfun(3)
```

Hint: Creating a table of values for *i*, *i/2-fix(i/2)*, *x* and *y* will be helpful. Also, it will be easier to check your work if you leave *y* in symbolic form until the end. In fact, I prefer that you state the answer in symbolic form. That is, including terms like $\sqrt{2}$ instead of 1.41421356....

Part C: The following problems require you to demonstrate how you can translate an algorithmic language to a programming language as well as building a Matlab m-file.

8. **[20 Points]:**

Up-Down algorithm is described as follow:

Let N be some positive integer.

If N is even, divide it by 2.

On the other hand, if it is odd, multiply it by 3 and add 1.

Repeat until N becomes 1.

It appears that this algorithm converges to 1 for all numbers N . Write a MATLAB sequence code to implement this algorithm. Your code should compute the number of iterations this algorithm will need to converge to one for any scalar number.

9. **[15 Points]:**

Design a function M-File that inputs the hyperbola parameters and outputs the value of the hyperbola function on a given interval.

10. **[15 Points]:**

Design a function M-File that performs the same as the *max* MATLAB built-in function