ES 314 Advanced Programming, Modeling and Simulation Fall 2012

Solutions to Mid-term # 1

Open book section

1) For each of the following assignment statements state whether or not it is correct. If correct, specify what value it would produce. If it is incorrect, just state "error". Assume a = 2, b = 90, c = –1, ma = [1 2; 2 3; 3 4] and mb = 1:4:10.

A) mb + [1, 2]

error

B) ma > 2

```
0     0
0     1
1     1
```

C) [a, [b, c]] == 90

```
[0, 1, 0]
```

D) a([a] + 1) = 5

```
[2, 0, 5]
```

E) mb == mb – 1

```
 [0, 0, 0]
```

F) b - 23 = c + 10

error

G) a([2:3:6]) = 4

```
[2, 4, 0, 0, 4]
```

H) [a, b, c] * [b, c, a]

error

2) Assume that the following statements have been executed:

```
>> x = 'hello';
>> y = 'there';
>> z = [x, y];
```

State the output that is produced by each of the following.

A: x == 1:5

```
[0, 0, 0, 0, 0]
```

B: length(z)

```
10
```

C: strrep(z,'ot','')

```
hellhere
```

D: z(z=='e')

```
eee
```

E: z([1:3 , 7:9])

```
hellher
```

3) Write a one-line statement in Matlab to perform the following.

A) compute θ in degrees for which sin(θ) = 0.15

```
180*asin(0.15)/pi
```

B) create a vector of length 100 all of whose entries are 5.

```
5 * ones(1, 100)
```

C) test if a vector A has at least one non-zero entry.

```
any(A ~= 0)
```

D) determine if the first and the second rows of a matrix A are identical.

```
all(A(1:) == A(2:))
```

E) generate a random real number between 1 and 3.

```
1 + 2* rand()
```

4) The following sequence of commands is typed. Write the output that is printed after each of the commands. (The operations are cumulative.)

```
>> v=1:3
[1, 2, 3]

>> u=[v;v.^2;1+v]
1     2     3
1     4     9
2     3     4

>> sum(u(1,:).*u(3,:))
20

>> u([3,1],1)'
[2, 1]
```

5) Name the function to perform each of the following operations:

(a) to make the scale on the x and y-axis equal.

```
axis('equal')
```

(b) to copy an image 'image1.jpg' to an array A.

```
imread('image1.jpg')
```

(c) to find the inverse of a matrix B.

```
B^(-1)
```

(d) to get the integer part of a real number.

```
floor
```

(e) to concatenate two strings s1 and s2.

```
strcat(s1, s2)
```

6) Explain the following terms:

(a) What is numerical indexing? Give an example.

If A is an array, any set of array elements can be accessed (or modified)using a vector of indices v using the format A(v). This is called numerical indexing.

```
Example: >> A = [1, 4, 9, 8]
>> A([1 2 1 3])
ans =

[1 4 1 9]
```

(b) What is logical indexing? Give an example.

If A is an array, (some of) the array elements can be accessed (or modified)using a Boolean vector v of the size equal to the size of the array. This is called numerical indexing.

```
Example: >> A = [1, 4, 9, 8]
>> A([false false false true])
ans =

[8]
```

(c) What is call by value?

Let f be a function of one variable say x. If the function f is called with input variable y, what is passed when the code for f is run is a copy of variable y. This means, even if variable x is modified in f, this change will have no effect on y. In contrast, call by reference would result in actually using the memory location where y is stored as the variable x so any change made to x in f will change the variable y.

(d) State the difference between a cell array and an ordinary array. Give an example where cell array is needed.

A cell array is an array of pointers – in contrast to ordinary array that hold data directly. A collection of strings can't be stored in ordinary array since they get concatenated to form a single string. A cell array can be (should be) used for this purpose.

Open book section. Create a separate file containing the function or script for problem, create a zip file containing your solutions and name it <last_name>.zip and submit to through your moodle account.

1) Start with a positive integer n, and generate successive numbers by computing the sum of squares of the previous number. For example, starting with 34, we get $3^2 + 4^2 = 25$, next we get $2^2 + 5^2 = 29$, then $2^2 + 9^2 = 25= 85$ etc. Eventually the sequence repeats: 25, 29, 85, 89, 145, 42, 20, 4, 16, 37, 58, 89

   You are to write a function square_seq that takes as input the starting number (25 in the above example) and produce as output a vector that contains all the numbers generated by the above procedure and stop with the first repeating number.

   Another example:

```
>> square_seq(15)

ans =

    15    26    40    16    37    58    89   145    42    20
4    16
```

Solution:

```
function out = square_seq(n)
out = [n];
next = sumsq(n);
while ~any(next==out)
    out = [out, next];
    next = sumsq(next);
end
out = [out, next];


function sum = sumsq(n)
sum = 0;
while ~(n==0)
    dig = mod(n,10);
    n = floor(n/10);
    sum = sum + dig*dig;
end;
```

2) Write a script *draw* that draws the following geometric figures using plot function: the circle of radius 2 with center (3,4), the line L through (0,0) and (3,4) and the tangent to the circle at the point of intersection of the circle and line L. (Note that there are two points of intersection; choose the point closer to origin.)

Hint: you can find the intersection point by solving a quadratic equation. Use Matlab to solve the equation.

(submitted by Fabian)

```
r = 2;
center = [3,4];
theta = linspace(0,2*pi,100);
x = r*cos(theta)+3;
y = r*sin(theta)+4;

%line L through (0,0) and center
x1 = linspace(0,5.5,100);
L = (center(2)/center(1))*x1;

%tangent line to circle at intersection closest to origin.
theta1 = atan(4/3);

xInt = r*cos(theta1+pi)+3;
yInt = r*sin(theta1+pi)+4;
x2=linspace(0,5.5,100);
tan_line = -1*(xInt-center(1))/(yInt-center(2))*(x2-
xInt)+yInt;

plot(x,y,x1,L,x2,tan_line,3,4,'-o');
axis('equal');
```

3) A *Latin square* of order k is a k by k matrix in which the numbers from 1 to k occurs exactly once in each row and exactly once in each column. Further, all the numbers from 1 to k occur exactly once along both diagonals.
Write a function (named ***latin***) in Matlab that tests if a given matrix is a *Latin square*.

```
>> A

A =

     1     2     3     4
     3     4     1     2
     4     3     2     1
     2     1     4     3

>> latin(A)
```

```
ans =

   1
```

```
function out = latin(A)
% tests if A is a Latin square
% assume A is a square matrix
[n, m] = size(A);
if ~(n== m)
    out = 0; return;
end;
for j = 1:n
    if ~(sort(A(:,j)')== 1:n) out = 0; return;
    end;
end;
for j = 1:n
    if ~(sort(A(j,:))==1:n)
        out = 0; return;
    end;
end;
vec = [];
for j = 1:n
    vec = [vec,A(j,j)];
end;
if ~(sort(vec) == 1:n)
    out = 0; return;
end;
vec = [];
for j = 1:n
    vec = [vec, A(n-j+1,j)];
end;
if ~(sort(vec) == 1:n)
    out = 0; return;
end;
out = 1;
```

4) Write a function ***move*** in Matlab that takes as input two vectors v and u, where v is a vector of n integers and u is a permutation of 1:n, and return a vector by moving the elements of v according to the permutation. Specifically, v(1) should move to position u(1), v(2) should move to position u(2) etc.

```
function out = move(u, v)
out(u) = v
```