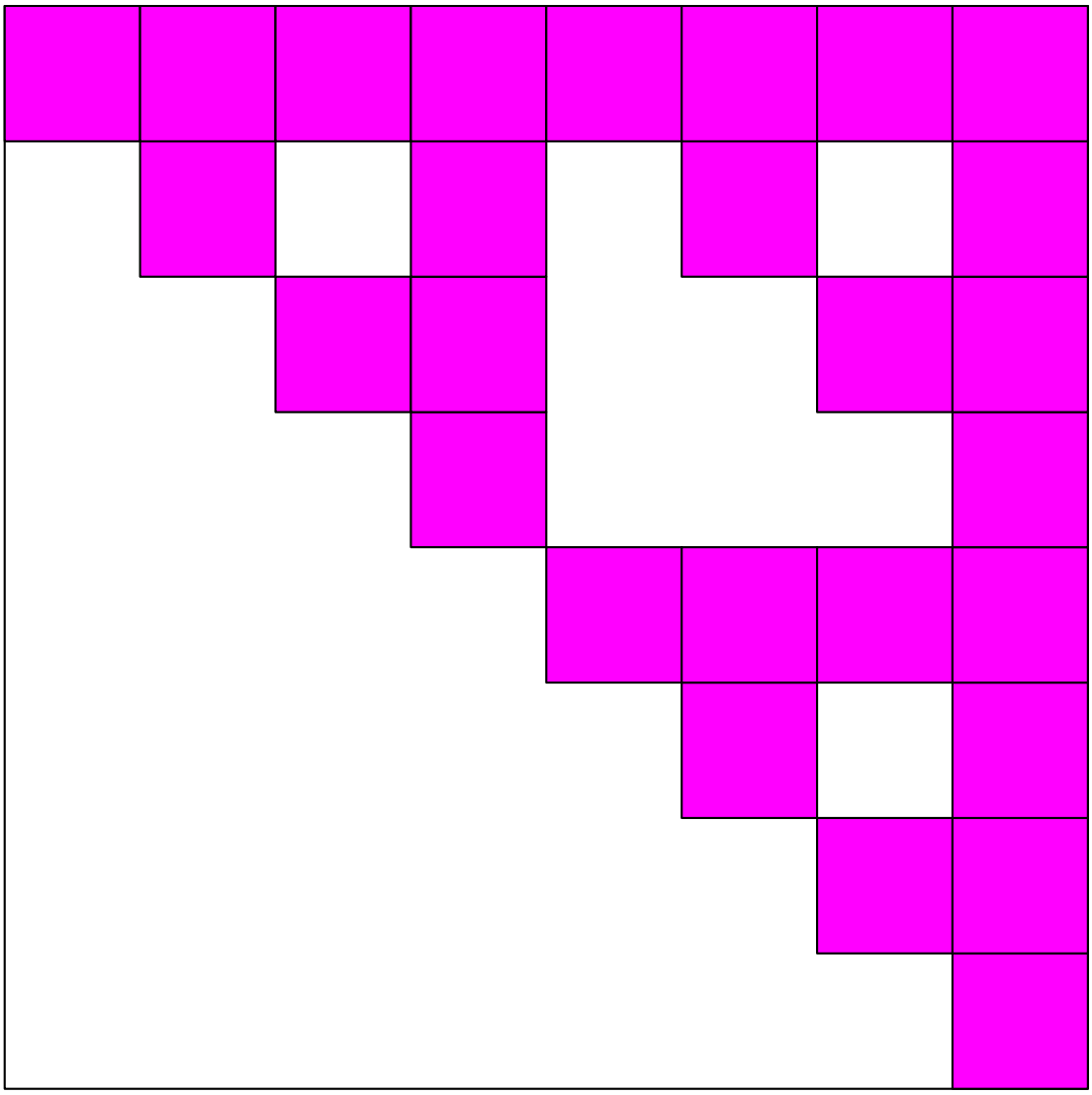
CS 1115: Final Exam

December 7, 2012

SOLUTION GUIDE



CS 1115: Final Exam

December 7, 2012

SOLUTION GUIDE

Problem 1	15 points	
Problem 2	15 points	
Problem 3	15 points	
Problem 4	10 points	
Problem 5	15 points	
Problem 6	15 points	
Problem 7	15 points	

85.5

Median = 87, Letter Grade Approximations: A: 90-100, B: 75-85 , C: 60-65

1. In this problem you are not allowed to use any MATLAB trig function or inverse trig function. Recall the trigonometric identities

$$\begin{aligned}
\sin(a+b) &= \sin(a)\cos(b)+\cos(a)\sin(b) \\
\cos(a+b) &= \cos(a)\cos(b)-\sin(a)\sin(b) \\
\sin(a/2) &= \sqrt{(1-\cos(a))/2} \qquad 0 \le a \le \pi/2 \\
\cos(a/2) &= \sqrt{(1+\cos(a))/2} \qquad 0 \le a \le \pi/2 \\
\sin(-a) &= -\sin(a) \\
\cos(-a) &= \cos(a)
\end{aligned}$$

Assume that the variables `s` and `c` contain the sine and cosine of some angle θ that satisfies $0 < \theta < \pi/2$.

(a) Write a script that assigns $\sin(3\theta/4)$ to `s2` and $\cos(3\theta/4)$ to `c2`. Use the fact that $\sin(0) = 0$ and $\cos(0) = 1$,

`6 points:`

```
v = sqrt((1+c)/2);                  % v = cos(theta/2)
u = -sqrt((1+v)/2);                 % u = -sin(theta/4)
v = sqrt((1+v)/2);                  % v = cos(-theta/4)
s2 = c*u+s*v;                       % s2 = sin(theta-theta/4)
c2 = c*v-s*u;                       % c2 = cos(theta-theta/4)
```

(b) Write a script that assigns $\sin(\alpha\theta)$ to `st` and $\cos(\alpha\theta)$ to `ct` where

$$\alpha = \frac{2^t - 1}{2^t}$$

and it is assumed that `t` is an initialized positive integer. Your solution must not make use of arrays.

`9 points`

```
v = c;                      % v = cos(theta )
for k=1:t-1
   v = sqrt((1+v)/2);       % v = cos(theta/2^k)
end
u = -sqrt((1-v)/2)           % u = sin(-theta/2^t)
v = sqrt((1+v)/2)           % v = cos(-theta/2^t)
st = c*u+s*v;               % st = sin(theta - theta-theta/2^t)
ct = c*v-s*u;               % ct = cos(theta - theta-theta/2^t
```

This problem is about turning mathbook formulas into algorithms.

2. Assume that the following fragment plays a message:

```
[v,f] = wavread('MyMessage);
sound(v,f)
```

Assume that `n` is a positive integer. Write a MATLAB fragment that computes a vector `y` such that

```
sound(y,f)
```

plays the message n times with $n-1$ intervals of silence in between the repetitions. The kth interval of silence should last approximately $T/2^k$ seconds where T is the duration of `sound(v,f)`.

```
y = v;
m = length(v);              % Playing a sound vector this long takes T seconds
for k=1:n-1
  z = zeros(ceil(m/2^k),1)  % A "silence vector" for duration T/2^k
  y = [y ; z ; v];          % Append a silence vector and a message vector
end
```

-2 or -3 for small mistakes liking using f incorrectly to determine the silence vector

3. Complete the following function so that it performs as specified:

```
 function A = CellMax(C)
% C is a length-p cell array of m-by-n matrices
% A is an m-by-n matrix and A(i,j) is the maximum of the (i,j)
% entries in C{1},...,C{p}.
```

Thus, if

```
C = { [ 1 2 ; 3 9 ; 5 6] , [ -1 3 ; 4 -1 ; 6 7]}
```

then

```
 A = [ 1 3; 4 9 ; 6 7]
```

```
p = length(C);
[m,n] = size(C{1};
A = zeros(m,n);
for i=1:m
   for j=1:n
       v = [];
       for k = 1:p
          v = [v C{k}(i,j)];
       end
       A(i,j) = max(v);
   end
end
```

```
MAX    Largest component.
       For vectors, MAX(X) is the largest element in X.
```

4. Assume the availability of the the following class:

```
classdef Point < handle
% A point has an x and y coordinate
  properties
     x
     y
  end

  methods
     function P = Point(x,y)
        P.x = x;
        P.y = y;
     end

     function d = Dist(ThisPoint,Q)
         % The distance from ThisPoint to the point referenced by Q.
         ...
     end
   end
end
```

(a) Would this be a way to display the coordinates of a point that is midway between (0,0) and (a, b) assuming that `a` and `b` are initialized? Explain.

```
P = Point(a,b);
Q = P/2;
Qx = Q.x
Qy = Q.y
```

1pt: must say that this is an errort because P is a reference and as such, it makes no sense in dividing it by two.

(b) Write an instance method that can be used to compute a point object that encodes the point that is midway between two given points.

2 points:

```
function Q = MidPoint(ThisPoint,P)
Q = Point((ThisPoint.x+P.x/2,(ThisPoint.y+P.y)/2)
```

(c) Assume the availability of the `Point` class above and

```
classdef CirclePolygon  < handle

% A CirclePolygon object represents a polygon whose vertices are on a circle.

properties
    C = Point.empty();  % The center of the circle
    r                   % The radius of the circle
    V = Point.empty();  % A vector of points that encodes the vertices
end

methods

      function P = CirclePolygon(C,r,theta)
    % C is a point, r is a positive real number, and theta is a length-n vector
    %      with the property that 0<=theta(1)< theta(2)<...< theta(n) < 2*pi.
    % P is a circle polygon with center C , radius r
    % and vertices (C.x + r*cos(theta(k)),C.y + r*sin(theta(k))) for k=1:n.
        ...
end
end
```

Write an instance method for this class that can be used to compute the perimeter of the polygon obtained by connecting the midpoints of a given circle polygon. You must take full advantage of the the `Point` class to receive full credit.

7 points

```
function d = Perimeter(ThisCirclePolygon)
P = ThisCirclePolygon;
% First, get all the midpoints...
n = length(P.V);
for k=1:n-1
   M(k) = P.V(k).Midpoint(P.V(k+1));
end
M(n) = P.V(n).Midpoint(P.V(1));
% Now the perimeter...
d = 0;
for k=1:n
    if k<n
        d = d + M(k).Dist(M(k+1));
    else
        d = d + M(k).Dist(M(1));
    end
 end
```

3 points for correctly using Midpoint and 3 points for correctly using Dist

5. (a) Assume that the availability of

```
function A = AreaCodes()
% A is a 500-by-3 character array that encodes all valid 3-digit area codes.
```

Develop an efficient implementation of the following function:

```
function alfa = IsAC(s)
% s is a length-3 string
% alfa is 1 if s encodes a valid area code and 0 if it is not.
```

8 points

```
A = AreaCodes();
k = 1;
while (k<=500 && ~strcmp(s,A(k,:))
   k = k+1;
end
alfa = k<=500
```

-4 for using a for-loop
-3 for or instead of and
-2 if you used three strcmp's

(b) A string `s` encodes a valid telephone number if it equals `'911'`, or if it has the form `'1xxxyyyzzzz'` where `xxx` is a valid area code and `yyy` is not a valid area code, or if it has the form `'yyyzzzz'` where `'yyy'` is not a valid area code. Otherwise, `s` does not encode a valid telephone number. Complete the following function so that it performs as specified.

7 points...

```
function alfa = IsNumber(s)
% s is a string of digit characters.
% alfa is 1 if s encodes a valid telephone number and is 0 otherwise

if length(s) == 3
   alfa = strcmp(s,'911');
elseif length(s) == 11
   alfa = strcmp(s(1),'1') && IsAC(s(2:4)) && ~IsAC(s(5:8))
elseif length(s) == 7
   alfa = ~IsAC(s(1:3))
else
   alfa = 0;
end
```

You may assume the availability of the function `ISAC` from part (a).

```
STRCMP Compare strings.
    TF = STRCMP(S1,S2) compares the strings S1 and S2 and returns logical 1
    (true) if they are identical, and returns logical 0 (false) otherwise.
```

6.(a) Assume the availability of the following function:

```
function DrawSquares(a,b,s,n)
if n==0
    fill([a a+s a+s a],[b b b+s b+s],'m')
else
    s = s/2;
    DrawSquares(a+s,b,s,n-1)
    DrawSquares(a,b+s,s,n-1)
    DrawSquares(a+s,b+s,s,n-1)
end
```

Sketch the output that would result if the following script is run:

```
plot([0 1 1 0 0],[0 0 1 1 0],'k')
hold on
DrawSquares(0,0,1,3)
```

Your sketch should shade those tiles that are magenta and leave white those tiles that are not magenta.

10 points

See Front Cover!

(b) Assume that `n` is a positive integer. As a function of n, what fraction of the unit square is colored magenta by the command `DrawSquares(0,0,1,n)`? Briefly explain your reasoning.

5 points

$(3/4)^n$ The magenta fraction at a given level is 3/4 of the magenta fraction of the previous level.

7. We say that an integer is a "good" integer if it has the form $2^i 3^j$ for some $i \geq 0$ and $j \geq 0$. Thus, $2^0 3^0 = 1$ and $2^5 3^4 = 32 \cdot 81 = 2592$ are examples. Assume that `M` is an initialized variable with positive value M. Write an efficient MATLAB script that assigns to `G` the number of good integers that are strictly less than M. Thus, if $M = 50$, then the value of `G` should be 15 since the numbers

$$
\begin{array}{llll}
2^0 3^0 & 2^0 3^1 & 2^0 3^2 & 2^0 3^3 \\
2^1 3^0 & 2^1 3^1 & 2^1 3^2 & \\
2^2 3^0 & 2^2 3^1 & 2^2 3^2 & \\
2^3 3^0 & 2^3 3^1 & & \\
2^4 3^0 & 2^4 3^1 & & \\
2^5 3^0 & & &
\end{array}
$$

are all less than 50. Your are not allowed to use any built-in functions in your solution, e.g., `log`. For your information, if $2^{i_1} 3^{j_1} = 2^{i_2} 3^{j_2}$, then $i_1 = j_1$ and $i_2 = j_2$.

```
  G = 0;
  i = 0;
  while 2^i < M
     j=0
     m = 2^i;
     while m* 3^j < M
        G = G+1;
        j = j+1;
     end
     i = i+1;
  end
end
```