ID (e.g., ab1234): _____ Name: _____ Expected Course Grade: _____

BIOMED 201 - Programming & Modeling for BME

Midterm Exam, 2013.05.07, Instructor: Ahmet Sacan

Sign the honor code below. **No credit will be given for the exam without a signed pledge.**

I have neither given nor received aid on this examination.

Signed: _____

There are 12 questions in this exam. Turn in Questions 1-8 before you start working on Questions 9-12. You can choose between questions 9 and 10; and choose between questions 11 and 12. Submit your programs for Questions 9-12 on ProgrammingBank, in addition to turning in your paper exam. Sign off your submission before exiting the room.

Q1 (5 pts). *Indexing.* Let **A** be a square matrix with **R** rows. Write a single statement that will set the antidiagonal elements of A to zero. Antidiagonal elements of A are A(R,1), A(R-1,2), ..., A(2,R-1), A(1,R). Do not use loops.

Q2 (5 pts). *Creating vectors.* Fill in the blanks below with what Matlab would display for the given expression.

```
>> [[7:-2:3] reshape(repmat([3 4],2,1),1,[])]
ans =

_____
```

Q3 (5 pts). *Binary and hexadecimal numbers*. In the first box below, write down the binary representation of the decimal number **170**. In the second box, write down the hexadecimal representation of the decimal number **170**.

```
0x
```

```
function y = apple( x )
global a;
persistent b;
if isempty(a); a=2; end
if isempty(b); b=3; end
a=a*2;
b=b*3;
y = a + b + x;
```

Q4 (5 pts). *Variable scope.* Fill in the blanks in the output.

```
>> x = apple( 5 );
>> global a; a = 7;
>> y = apple( 9 );
>> [x y a]
ans =

_____________________________________
```

Q5 (5 pts). *Indexing.* Let **M** be a matrix with **1000** rows and **2000** columns. **M(12345)** can equivalently be expressed as **M(x,y)**. What are the values of **x** and **y**?

x=_____, y=_____

Q6 (5 pts). *Vectorizing code, element-wise operations, logical indexing.* Let **X,Y,Z,M** be column matrices with the same size. Rewrite the for loops below without any using loops. You may define additional variables.

```
M=zeros(0,3);
for i=1:20
 for j=1:20
  for k=1:20
   if i*i+j*j==k*k
    M(end+1,:)=[i j k];
   end
  end; end; end
```

```
[X,Y,Z]=meshgrid(1:20,1:20,1:20);
```

Q7 (10 pts). *Nested for loops.* Fill in the blanks in the output below.

```
>> x = zeros(2,0);
>> for a = [ 3 5 ]
>>  for b = a*10:a*10+1
>>    x([2 1],end+1:end+2) = [ a b; a+1 b+1];
>>  end; end
>> disp ( x )

_____________________________________________________
```

Q8 (10 pts). *Operator precedence, logic.* Fill in the blanks in the outputs below.

ID (e.g., ab1234): _____ Name: _____ Expected Course Grade: _____

```
>> disp( -5 >= -3 >= 0 )
```

_____

```
>> v=[5 10 -10 10-10 ];
>> disp([any(v) all(v) numel(v)])
```

_____

```
>> disp( 8 < 9 - 2 )
```

_____

```
>> X=randi(2,1,10)-1;
>> disp(mean(double((X & ~X)|(X | ~X))))
```

_____

ID (e.g., ab1234): _____ Name: _____ Expected Course Grade: _____

I have neither given nor received aid on this examination.

Time of submission: _____

Signed: _____

You may attempt both Q9 and Q10. The one with the higher grade will be counted for your midterm grade.

Q9 (30 pts). *Loops or vectorized code*. Write a function **getatriangle(R, f,b)** that takes the number of rows **R** and returns a character matrix that has the shape of a triangle filled with the character **f** on a background filled with the character **b**. Assume that **R** is a positive integer and that **f** and **b** are single characters. If **b** is not given, use '**.**'. If **f** is not given, use '**#**'.

```
>> getatriangle(4,'@','-')
ans =

---@---
--@@@--
-@@@@@-
@@@@@@@
```

```
>> getatriangle(5)
ans =

....#....
...###...
..#####..
.#######.
#########
```

Q10 (30 pts). *Logical indexing, loops.* Comma separated values (CSV) is text file format commonly used for data exchange. In this problem, you are asked to parse a value from a CSV string. Consider the following matlab string s, containing 6 values:

```
s='10,"Glycogen phosphorylase, liver",PYGL,183353,"human,Homo sapiens","DB00114,DB00131"'
```

Notice that some of the values can contain a comma; such values are enclosed with quotes. Write a function **parsecsvline(s,n)** that takes a CSV string **s** and returns its **n**th value. If the value is enclosed in quotes, you mush strip off the quote characters. If **s** contains less than **n** values, return an empty string. Do not use Matlab's csvimport(), csvread(), or xlsread() function.

```
>> s='10,"Glycogen phosphorylase, liver",PYGL,183353,"human,Homo sapiens","DB00114,DB00131"';
>> disp(parsecsvline(s,1))
10
>> disp(parsecsvline(s,2))
Glycogen phosphorylase, liver
>> disp(parsecsvline(s,3))
PYGL
>> disp(parsecsvline(s,5))
human,Homo sapiens
>> disp(['#' parsecsvline(s,10) '#'])
##
```

ID (e.g., ab1234): _____ Name: _____ Expected Course Grade: _____

You may attempt both Q11 and Q12. The one with the higher grade will be counted for your midterm grade.

Q11 (20 pts). *Loops & selection; or vectorized code & logical indexing.* Write a function **turkey1994(N)** that takes an integer **N** and returns a 2-column matrix where each row contains integers **a,b** where $1 \leq a < b \leq N$ and $a^2 + b^2 + 3$ is divisible by ab.

```
>> disp(turkey1994(20))
     1     2
     1     4
     2     7
     4    19
```

Q12 (20 pts). *Mathematical expressions.* Sigmoid function has found use in a number of statistical regression and mathematical modeling applications. Consider the sigmoid function defined by:

$$f(x) = \frac{1}{1 + e^{-kx}}$$

where **x** is real and **k** is a positive constant. The value of **f(x)** can be approximated by making use of the following formula:

$$e^t = \lim_{n \to \infty} \left(1 + \frac{t}{n}\right)^n$$

In this question, you will write a function **sigmoidlike(x,k,n)** that approximates **f(x)** using:

$$f(x) \approx \frac{1}{1 + \left(1 - \frac{kx}{n}\right)^n}$$

If **x** is a matrix, the function output should be a matrix of the same size, where each element is the sigmoid approximation the corresponding input element.

```
>> disp(sigmoidlike(0.5,4,5))
    0.9279
>> disp(sigmoidlike(0.5,4,20))
    0.8916
>> disp(sigmoidlike([-1:.5:1],2,10))
    0.1390    0.2783    0.5000    0.7415    0.9030
```