Drexel-Email-ID: _____ Name: _____ Expected Course Grade: _____

BIOMED 201 - Programming & Modeling for BME

Final Exam, 2012.12.13, 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 10 questions in this exam. Turn in Questions 1-7 before you start working on Questions 8-10. Submit your programs for Questions 8-10 on ProgrammingBank, in addition to turning in your paper exam. Submit only one of the Questions 9 and 10.

Q1 (5 pts). *Indexing.* Random sampling is an important task in statistical data analysis. Let **A** be a non-empty matrix where each row represents a sample and each column represents a different type of measurement. Write a single statement that will assign into **B**, 1000 samples randomly picked from **A**. Some samples of **A** may appear multiple times in **B**. Your code should work for any sized matrix **A**. Do not use loops. Hint: The function **randi(p,q,s)** generates a matrix of **q** rows and **s** columns with each element being a random integer between 1 and **p**.

```
B = A(randi(size(A,1),1000,1), :)
```

Q2 (5 pts). *Linear Indexing.* Let **M** be a matrix with **100** rows and **200** columns. **M(12, 34)** can equivalently be expressed as **M(x)**. What is the value of **x**?

```
3312
```

Q3 (5 pts). *Logical Indexing.* Let **A** be a linear vector any size, containing integers. Write a single statement that will assign into B, all odd elements of A. e.g., if A is [2 4 5 5 6 3 -1 1], B will be [5 5 3 -1 1]. Your code should work for any sized vector A. Do not use loops.

```
B = A( mod(A,2)==1 )
```

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

```
>> points = struct( 'a', 10, 'b',{3 4 7}, 'c',{1 2 5} );
>> disp( [ points.a points.(char('b'+1)) ] )

    10    10    10     1     2     5
```

Drexel-Email-ID: _____ Name: _____ Expected Course Grade: _____

Q5 (10 pts). *cells*. Fill in the outputs below.

```
>> m = {3:7 {[3:7 2:9]}; {3:7 2:9} {{3:7} {2:9}}; {} [] };
```

```
>> size(m)
     3     2

>> size( m(3) )
     1     1

>> size( m{3} )
     0     0
```

```
>> m{4}{1}(2)
     4

>> m{5}{2}{1}(2)
     3
```

Q6 (10 pts). *Vectorized code*. Write a single statement to calculate approximation of π using the following formula. Assume that variable **n** has already been assigned a positive integer value. Do not use loops.

$$\frac{\pi^2}{6} \approx 1+\frac{1}{4}+\frac{1}{9}+\frac{1}{16}+\cdots+\frac{1}{n^2}$$

```
>> eulerpiapprox = sqrt( 6* sum( 1./((1:n).^2) ) )
```

Q7 (20 pts). *string functions, indexing*. Fill in the outputs below.

```
>> s='ATOM..126..NH1.ARG.A..17'
>> disp( s(strfind(s,'NH1')+3:strfind(s,'17')-1) )
.ARG.A..

>> variables = {'gender', 'height', 'weight', 'systolic', 'diastolic'};
>> data=[0 63 130 70 110; 1 70 160 85 125; 0 65 140 82 115; 1 68 180 85 130];
```

```
>> I=strcmpi(variables, 'WEIGHT')
     0     0     1     0     0

>> ind=find(I)
     3

>> data(:, I)'
   130   160   140   180

>> data(:,ind)'
   130   160   140   180
```

```
>> J=data(:,1)==1;
>> find(J)'
     2     4

>> data(J,I)'
   160   180


>> data(~J,I)'
   130   140
```

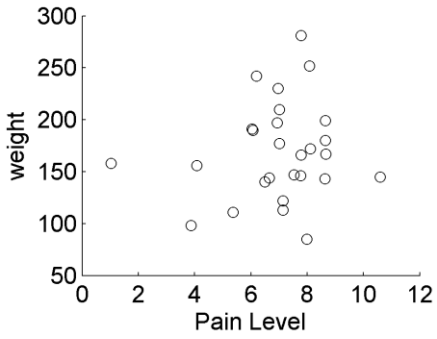
Drexel-Email-ID: _____ Name: _____ Expected Course Grade: _____

I have neither given nor received aid on this examination.

Time of submission: _____

Signed: _____

Q8 (20 pts). *data analysis*. Write a Matlab function **crps_femalepatientpaincorr(factor)**, that takes a string input *factor* and plots the Pain levels versus the values for the factor under consideration, only in female CRPS patients (the example on the right shows the plot when factor is the Weight). The function should then return the correlation of that factor with Pain, in female CRPS patients. Use the crps_final.zip file provided online for the data and helper functions. Your submission should not require any other non-matlab function than what is provided in crps_final.zip.




```
>> disp( crps_femalepatientpaincorr ( 'weight' ) )
    0.1451
```

Q9 (20 pts). *strings, file IO*. In this problem, you will work with the SCOP hierarchical database for classification of protein structures, to identify the structural family of a given protein domain. Download the SCOP file from http://scop.berkeley.edu/downloads/parse/dir.des.scop.1.75A.txt into your current working directory. Write a function **scop_domain2family(domain)** that takes a 7-character domain identifier and returns the family name for that domain.

SCOP file is a tab-delimited file where each line contains: *unique identifier* (numeric), *row-type* (2 characters), *hierarchical-code* (e.g., a.1 or a.1.1 or a.1.1.1), *domain name* (e.g.,'d1ux8a_'), and *description*. You need to first find the line where the *row-type* is 'px' and the *domain* matches the input to your function; and extract the *hierarchical-code* from that row. You then need to find the line where the *row-type* is 'fa' and the *hierarchical-code* matches the one for your *domain*; and extract the *description*. This *description* is what you return from your function. E.g., for *domain* 'd1ux8a_', you first extract its *hierarchical-code* 'a.1.1.1', then extract the name for that family: 'Truncated hemoglobin'.

```
>> disp( scop_domain2family('d1ux8a_') )
Truncated hemoglobin

>> disp( scop_domain2family('d1vhba_') )
Globins

>> disp( scop_domain2family('d1uerd1') )
Fe,Mn superoxide dismutase (SOD), N-terminal domain
```

Q10 (20 pts, graded only if you do not submit Q9). *Strings, plotting, loops, vectorized code.* Reaction and diffusion of a few substances (morphogens) can generate a vast number of patterns seen in developmental biology. In this problem, we will consider a 2-dimensional arrangement of N^2 cells arranged on an **N**-by-**N** grid and the reaction-diffusion of 2 morphogens **A** and **B**. You will simulate the reaction-diffusion of **A** and **B** over **T** discrete time steps.

Concentrations of A and B at each time point are calculated from their concentrations at previous time point. Furthermore, the concentrations in any cell depend on the concentrations in its neighbor cells. The change in concentrations of A and B are described by the following formulas:

$$\Delta a_{i,j} = D_a\left(a_{i+1,j}+a_{i-1,j} + a_{i,j+1}+a_{i,j-1} - 4a_{i,j}\right) + k\left(16 - a_{i,j}b_{i,j}\right)$$

$$\Delta b_{i,j} = D_b\left(b_{i+1,j}+b_{i-1,j} + b_{i,j+1}+b_{i,j-1} - 4b_{i,j}\right) + k\left(a_{i,j}b_{i,j} - b_{i,j} - 12 - \beta_{i,j}\right)$$

where $a_{i,j}$, $b_{i,j}$, and $\beta_{i,j}$ are the concentrations of the morphogens A and B and a substrate β in the cell located in the ith row and jth column of the grid, D_a and D_b are the diffusion rates of the morphogens, and k is the reaction rate. Assume that periodic boundary conditions are used to update the cells in the first and last rows and columns, e.g., in the above formulas, for the first row, $a_{i-1,j}$ would refer to $a_{N,j}$ and for the last row, $a_{i+1,j}$ would refer to $a_{1,j}$.

Write a function **reaxdiff2d.m** that takes **N** and **T** as input and simulates the reaction-diffusion of A and B. Use the default values $N = 200$ and $T = 2000$, when these inputs are not provided to your function. Use the following values for constants: $D_a = 0.2, D_b = 0.02,\ k = 0.005$. The initial values of $a_{i,j}$ and $b_{i,j}$ in all of the cells are 4.0. The initial values of $\beta_{i,j}$ should be random numbers between -0.1 and +0.1. When applying the above formulas to update concentrations of A and B, you need to enforce that concentrations of A and B never go below zero in any cell (i.e., if the above formulas cause the concentration of A or B to go below zero in a cell, you need to manually set that concentration to zero.)

Your function should plot the final concentrations of A and B on the same figure as two subplots, as shown in the following example below. You can use the imagesc() function to show the values in a matrix as an image. Note that, since the values of β_i are set randomly, your simulation may provide a different pattern than shown below.