

BIOMED 201 - Programming & Modeling for BME

Final Exam, 2013.06.06, 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 two of the Questions 8-10.

Q1 (5 pts). *Indexing.* Random sampling is an important task in statistical data analysis. Let **A** be a non-empty matrix with more than 1000 rows, where each row represents a sample and each column represents a different type of measurement. In at most two statements, assign into **B**, 1000 samples randomly picked from **A**. Samples of **A** should not appear multiple times in **B**. Your code should work for any sized matrix **A**. Do not use loops. Hint: The function **p=randperm(n)** returns a random permutation of integers from 1 to **n** inclusive; e.g., **randperm(6)** will return numbers 1 to 6 in a random order, e.g., it may return [5 2 6 4 1 3].

```
I=randperm(size(A,1))
```

```
B = A(I(1:1000),:)
```

Q2 (5 pts). *Linear Indexing.* Let **M** be a matrix with 100 rows and 200 columns. **M(19,19)** can equivalently be expressed as **M(x)**. What is the value of **x**?

```
1819
```

Q3 (5 pts). *Logical Indexing.* Let **A** be a linear vector any size, containing positive integers. Write a single statement that will assign into **B**, all odd elements of **A** that are also squares of integers. e.g., if **A** is [2 9 4 5 25 16], **B** will be [9 25]. Your code should work for any sized vector **A**. Do not use loops.

```
B = A(mod(A,2)==1 & mod(sqrt(A),1)==0)
```

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

```
>> points = struct('a',{10 20 30}}, 'b',{3 4 7}}, 'c',{1 2 5}});
>> disp([points(2).a{1} points(2).b points.(char('c'-1))])
```

```
10 4 3 4 7
```

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

<pre>>> m = {3:7 {[3:7 2:9]} {3:7 2:9} ; {{3:7} {2:9}} {} [] };</pre>	
<pre>>> size(m) 2 3 >> size(m(1:4)) 1 4 >> size(m{2,2}) 0 0</pre>	<pre>>> size(m(4)) 1 1 >> m{2}{2}{1}(2) 3 >> m{3}{1}(2:3) 4 5</pre>

Q6 (10 pts). *Vectorized code.* Write a single statement to calculate an approximation of $\ln(x)$ and assign it into variable `lnx` using the following formula. Assume that variable `n` has already been assigned an even number, and `x` is a double number where $|x - 1| \leq 1$ and $x \neq 0$. Do not use loops.

$$\ln(x) \approx (x - 1) - \frac{(x - 1)^2}{2} + \frac{(x - 1)^3}{3} - \frac{(x - 1)^4}{4} \dots + \frac{(x - 1)^{n-1}}{n-1} - \frac{(x - 1)^n}{n}$$

<pre>>> lnx = sum((x-1).^(1:n)./(1:n).*repmat([1 -1],1,n/2))</pre>
--

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

<pre>>> s='46460,dm,a.1.1.1,-,Protozoan/bacterial_hemoglobin'; >> vars = {'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];</pre>	
<pre>>> a=find(s==' '); >> disp(s(a(2)+1:a(3)-1)) a.1.1.1 >> b=strfind(s,'/'); >> c= strfind(s,'hem'); >> disp(s(b+1:c-2)) bacterial >> I=strcmpi(vars, 'height') 0 1 0 0 0 >> data(:, I) ' 63 70 65 68</pre>	<pre>>> J=logical(data(:, ... strcmp(vars, 'gender'))); >> J=find(J) ' 2 4 >> data(J, I) ' 70 68 >> data(J, ~I) 1 160 85 125 1 180 85 130</pre>

I have neither given nor received aid on this examination.

Time of submission: _____

Signed: _____

Submit only two of the Questions 8-10. If you submit all three, your lowest two grades will be used!

Q8 (20 pts). *Vectorized code or loops.* Write a Matlab function **taylorlog(x,n)**, that takes a number x where $|x - 1| \leq 1$ and $x \neq 0$, and a positive integer n (n can be odd or even), and returns an approximation to $\ln(x)$ using the formula below. When n is not given, use $n = 10$.

$$\ln(x) \approx (x - 1) - \frac{(x - 1)^2}{2} + \frac{(x - 1)^3}{3} - \frac{(x - 1)^4}{4} \dots + \frac{(x - 1)^{n-1}}{n-1} - \frac{(x - 1)^n}{n}$$

```
>> disp(taylorlog(1.5,3))
    0.4167
>> disp(taylorlog(.5))
   -0.6931
```

Q9 (20 pts). *strings, file IO.* ID and name conversion is one of the common tasks in Bioinformatics. In this problem, you will write a function **symbol=geneidtosymbol(id)** that will return the symbol of a gene, given its GeneID. The GeneID to symbol conversion should be looked up from a file named "gene_info.txt". Download and use the file available from http://sacan.biomed.drexel.edu/ftp/bmes201/final.20123/gene_info.txt (which contains the first 100 lines of the file available from: ftp://ftp.ncbi.nlm.nih.gov/gene/DATA/gene_info.gz).

```
>> disp( geneidtosymbol (1246500) )
repA1
>> disp( geneidtosymbol (5961932) )
pMF1.12
>> disp( geneidtosymbol (1343060) )
orf17/2.9-9 lp
```

Q10 (20 pts). *Loops, plotting, simulation.* In order to investigate HIV infection and response to therapy, numerous differential equation models have been proposed. In this problem, you will implement and simulate one of these models. A widely adopted model to describe plasma viral loads in HIV infected individuals includes the following variables (De Leenheer, et.al., “Virus dynamics: a global analysis”, SIAM J. Appl. Math., 2003):

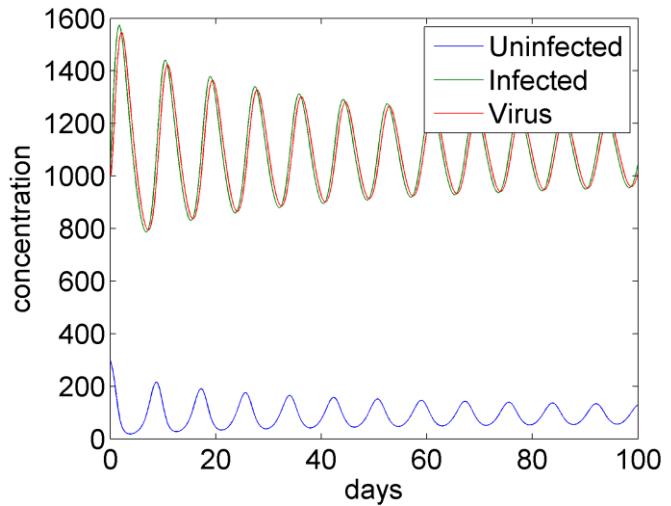
- U: Concentration of uninfected cells
- I: Concentration of infected cells
- V: Concentration of free virus particles in blood.

The change in the concentrations of these variables is modeled by the following differential equations:

- $\Delta U = f(U) - kVU$, where $f(U) = \sigma - \alpha U + pU(1 - \frac{U}{U_{max}})$
- $\Delta I = -\beta I + kVU$
- $\Delta V = -\gamma V + N\beta I$

Write a function `hivsimulate(U,I,V,deltat,T)` that takes the initial concentrations **U**, **I**, and **V** and simulates the concentrations of U, I, V up to **T** days using a time step of **deltat**. Plot the concentrations against time as shown on the right and return a 4-column matrix that contains the time and concentrations of U, I, V across the entire simulation. Use the following parameter values:

$\sigma = 10\text{day}^{-1}\text{mm}^{-3}$, $\alpha = 0.02\text{day}^{-1}$, $p = 3\text{day}^{-1}$, $U_{max} = 1500\text{mm}^{-3}$, $\beta = 0.24\text{day}^{-1}$, $\gamma = 2.4\text{day}^{-1}$, $k = 0.0027\text{mm}^3\text{day}^{-1}$, and $N = 10$.



```
>> disp( hivsimulate ( 300, 1000, 1000, 0.1, 100 ) )
1.0e+03 *

      0      0.3000      1.0000      1.0000
0.0001      0.2914      1.0570      1.0000
0.0002      0.2836      1.1103      1.0137
0.0003      0.2754      1.1613      1.0369
0.0004      0.2662      1.2105      1.0667
0.0005      0.2557      1.2581      1.1012
...
[truncated long output]
...
0.0995      0.1211      1.0002      0.9781
0.0996      0.1232      1.0081      0.9834
0.0997      0.1252      1.0167      0.9893
0.0998      0.1269      1.0257      0.9959
0.0999      0.1284      1.0352      1.0030
0.1000      0.1296      1.0452      1.0108
```