

BIOMED 201 - Programming & Modeling for BME

Final Exam, 2013.12.04, 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 **7 questions** in this exam. Turn in your paper exam before you start working on Questions 6-7. Submit your programs for Questions 6-7 on ProgrammingBank.

Q6 (25 pts). *Loops.* Similarity between proteins can be identified using “sequence alignment”, where a particular alignment can be modeled as traversal of a matrix that represents all pairwise alignments of the residues from two sequences. In this problem, you will find the score of an alignment by traversing a scoring matrix by a given path.

Write a function `traversescorematrix(mat,pat)`, where `mat` is a numerical matrix and `pat` encodes the path to be taken on that matrix, and returns the **score** corresponding to the path. A path contains a series of steps that starts from the top-left corner of the matrix and ends at the bottom-right corner. Each step in the path can be one of right, down, or diagonal moves, encoded by numbers 1, 2, and 3, respectively. The score of a path is defined as the sum of the values in the elements of the matrix traversed by the path. The example on the right shows a 5-by-6 scoring matrix and the path shown by the arrows is: [3, 1, 2, 2, 3, 1, 1], and the corresponding score is: $4+2+6+5+4+8-3-1 = 17$.

4	-2	6	0	6	5
-4	2	6	3	10	7
2	-3	5	5	3	2
0	4	-4	2	0	-3
-2	3	-3	8	-3	-1

```
>> disp( traversescorematrix([4 -2 6 0 6 5; -4 2 6 3 10 7; 2 -3 5 5 3 2; ...  
                             0 4 -4 2 0 -3; -2 3 -3 8 -3 -1], [3, 1, 2, 2, 3, 1, 1] ) )  
  
17
```

Q7 (25 pts). *strings, file IO.* In this problem, you will write a function `site=getsite(enzyme,rebasefile)` that takes a restriction enzyme name and returns its recognition site, as identified from a file containing information about restriction enzymes. If a rebase file is not provided, use 'rebase_small.txt'.

Download and use the example rebase file available from

http://sacan.biomed.drexel.edu/ftp/bmes201/final.20131/rebase_small.txt The format of the rebase file is self-descriptive.

```
>> disp( getsite ('AaaI','rebase_small.txt') )  
C^GGCCG  
>> disp( getsite ('AamI','rebase_small.txt') )  
?  
>> disp( getsite ('AbaUI') )  
C(11/9)
```

Q1 (10 pts). *Binary numbers and base conversion.* Write the decimal number **92** in binary and in base-3.

binary: 1011100	In base 3: 10102
------------------------	-------------------------

Q2 (10 pts). *structs.* Fill in the blanks in the output.

```
>> p=struct('ice',[3 5],'cream',{20 40},'cake',{70 90});
>> ice='cream';
>> disp( [ p(1).ice p(1).cream p(1).cake{1} p(1).(ice) ] )
```

3 5 20 70 20

Q3 (10 pts). *cells.* Fill in the outputs below.

<pre>>> m={[1 5] {1 5}; 1:5 {1:5}};</pre>	
<pre>>> disp([size(m(1)) size(m(2)); ... size(m(3)) size(m(4))])</pre> <p style="text-align: center;">1 1 1 1 1 1 1 1</p>	<pre>>> disp([size(m{1}) size(m{2}); ... size(m{3}) size(m{4})])</pre> <p style="text-align: center;">1 2 1 5 1 2 1 1</p>

Q4 (10 pts). *Logical Indexing.* Let **A** be a cell array of student names and **B** be a numerical vector of student grades. Write a single statement that will assign into **C** the names of the students whose grade is 100. E.g., if **A** is {'Ahmet', 'Ashley','Berk','Cory','Derek','Emily'} and **B** is [90 100 100 85 99.9 100], then **C** should become {'Ashley', 'Berk', 'Emily'}. Assume **A** and **B** have the same number of elements. Do not use loops.

```
C = A(B==100)
```

Q5 (10 pts). *string functions, indexing.* Fill in the outputs below.

```
>> vars = {'group', 'height', 'weight'};
>> data=[1 60 130; 1 70 160; 2 64 140; 1 68 180; 3 65 150; 2 66 200];
>> disp( data(:, strcmp(vars,'height'))' )
```

60 70 64 68 65 66

```
>> disp( data( data(:,1)==1, strcmp(vars,'height'))' )
```

60 70 68

```
>> disp( data( data(:,strcmp(vars,'group'))==1, strcmp(vars,'height'))' )
```

60 70 68