

Name: _____ NetID: _____

It is a violation of the Code of Academic Integrity to look at any exam other than your own, to look at any reference material outside of this exam packet, to communicate with anyone other than the exam proctors, or to otherwise give or receive any unauthorized help during the exam.

Academic Integrity is expected of all students of Cornell University at all times. By submitting this exam, you declare that you will not give, use, or receive unauthorized aid during this examination.

Statement of integrity: *I have not, and will not, violate the rules of academic integrity on this exam.*

(Signature)

Instructions:

- Check that this packet has 8 double-sided sheets.
- This is a 90-minute, closed-book exam; no calculators are allowed.
- The exam is worth a total of 100 points, so it's about one point per minute!
- Read each problem completely, including any provided code, before starting it.
- Do not modify any *given* code unless asked to do so.
- Use the back of the pages if you need additional space.
- If something is truly ambiguous, you may raise your hand to ask for clarification. But try to use your best judgment.
- Clarity, conciseness, and good programming style count for credit.
- Indicate your final answer. If you supply multiple answers, you may receive a *zero* on that question.
- Use only MATLAB code. No credit for code written in other programming languages.
- Assume there will be no input errors.
- Do not use `switch`, `try`, `catch`, `break`, `continue`, or `return` statements.
- Do not use built-in functions that have not been discussed in the course.
- You may find the following MATLAB predefined functions useful: `abs`, `rem`, `min`, `max`, `zeros`, `ones`, `length`, `size`, `uint8`, `fprintf`, `disp`, `strcmp`, `str2double`, `fopen`, `feof`, `fgetl`, `fclose`, `sort`

Examples: `rem(5,2)` → 1, the remainder of 5 divided by 2
`zeros(1,4)` → 1 row 4 columns of zeros
`length([2 4 8])` → 3, length of a vector
`[nr,nc,np]=size(M)` → dimensions of M: nr rows, nc columns, np layers
`uint8(4.7)` → the integer (type `uint8`) value 5
`strcmp('cat','dogs')` → 0, the two strings are not identical
`str2double(' -2.6')` → -2.6, a type double scalar
`fid=fopen('a.txt','r')` → open a file for reading and return file ID
`feof(fid)` → true if no more lines are left to read in a file
`s=fgetl(fid)` → read next line in file and store string in `s`
`fclose(fid)` → close an open file
`[y,idx]=sort(x)` → elements of `x` sorted in ascending order returned in `y` with the property that `y(k)=x(idx(k))`

Question 1. (15 points)

(a) Assume that the following script has just been executed by MATLAB:

```
v = [212, 192];  
w = uint8(v);  
a = v(1) + v(2);  
b = w(2) - w(1);  
c = 2.0 * w(1);  
d = (v(1) + w(2)) / 4;  
e = v(1) > 200;
```

What are the values and types of the variables a through e, as would be seen in the MATLAB Workspace? Valid types are: double, uint8, logical, char, cell.

Variable	Value	Type
a		
b		
c		
d		
e		

(b) Fill in the value of the following size() expressions (your answer should be a short vector of integers):

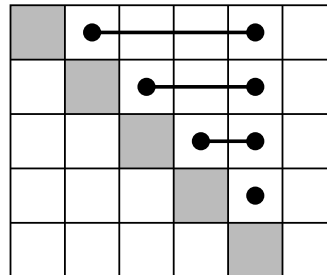
```
a= zeros(3, 5);  
k= 1;  
adims= size(a(2:3, k:k+2)) % -----  
  
b= zeros(240, 360, 3);  
m= 96;  
bdims= size(b(120, m-3:m+3, :)) % -----
```

Question 2. (10 points)

Fill in the placeholders [A] through [F] with variable names and expressions so that the code below performs a *row-major* traversal of the elements of matrix M above and to the right of (but excluding) the diagonal and excluding the last column, as illustrated below (do not make any assumptions about the size of M based on the illustration).

Hint: Write out the indices of the first few elements that are printed, in order, then look for a pattern. Consider matrices taller or wider than the illustration, and check edge cases.

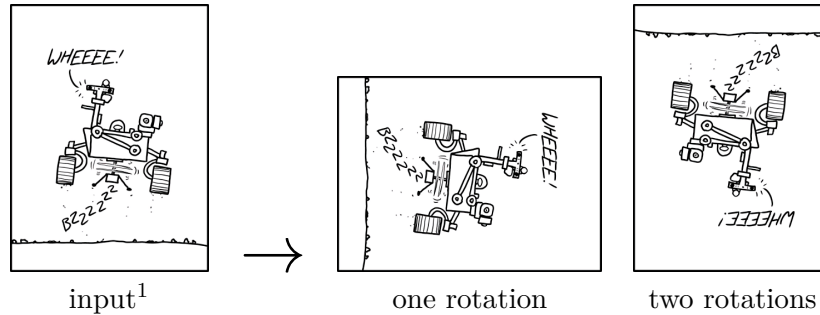
```
[nr, nc] = size(M);  
  
for [A] = [B]:[C]  
    for [D] = [E]:[F]  
        disp(M(r,c))  
    end  
end
```



- [A]: _____
- [B]: _____
- [C]: _____
- [D]: _____
- [E]: _____
- [F]: _____

Question 3. (20 points)

Declare and implement a function, `rotateImage`, to rotate an image clockwise by a multiple of 90° . The function should have two input arguments: the image to rotate (a 3D uint8 array) and the number of rotations to perform (an integer ≥ 0); it should return the rotated image (also a 3D uint8 array). For maximum points, make effective use of vectorized code and avoid redundant work. You do not need to write a specification comment.



Hint: Draw a picture of the input image, labeling the indices of a few pixels. Then draw the output after a single rotation and label the indices where those pixels ended up. Write a scalar formula for copying a single pixel before attempting a vectorized solution (a correct loop-based solution is better than an incorrect vectorized one).

¹Image from xkcd.com

Question 4. (25 points)

A monkey at a typewriter has filled a page with text, stored in a character matrix. We want to look for broken fragments of Shakespeare in this document. We proceed by *decomposing the problem*.

(a) The first step is to search for a single word in a single line (row) of text. **Implement the following function** to perform this search starting from a given column:

```
function c = findInLine(w, txt, c)
% Search for string `w` within character row vector `txt`, starting from
% column `c`. Return the column on which the first letter of the word was
% found, or 0 if the word is not present.
% Example: findInLine('be', 'toxbe', 3) is 4, but
%          findInLine('to', 'toxbe', 3) is 0 (because 'to' is not found at
%          or after column 3)
```

(b) Assume that a collaborator has completed the next step: searching for a single word on a page of text, starting from a given position. They have written function `findWord()` (using your `findInLine()` as a helper—thanks!) as specified below:

```
function [r, c] = findWord(w, txt, r0, c0)
% Search for word `w` in lines of character matrix `txt`, starting from
% column `c0` of row `r0`. Return the row `r` on which the word is found,
% and the column `c` containing its first letter, or return (r=0,c=0) if
% the word is not present.
% Edge cases: If c0 > # cols of txt, then search begins on the next line,
% at index (r+1,1). If r0 is 0, return (0,0) immediately.
```

The final step is to search for all words of a phrase, in order (but possibly separated), on the page. **Implement function `findPhrase()`**, specified below. Make effective use of (but do not implement!) function `findWord()`.

```
function f = findPhrase(p, txt)
% Search for each word of `p` (a 1D cell array of strings) in lines of 2D
% character matrix `txt`, requiring that each word occurs somewhere after
% the previous one (either later in the same row, or on a subsequent row;
% found words may not overlap). Return true if all words are found in
% order, false otherwise.
% Example: findPhrase({'to','be','or'}, ['ztoq';'beor']) is true,
%           findPhrase({'muse','of','fire'}, ['bmuseo';'fireof']) is false.
```

Question 5. (30 points) Consider student vaccination data stored in a file like the following:

```
2021-03-01 M 3141592 Cody Clay
2021-03-02 P 2718282 Mae Goodwin
2021-03-15 J 1618034 Hakim Umayya
2021-03-17 P 2718282 Mae Goodwin
```

The records are ordered by the date a vaccine dose was received, which occupies the first 10 columns of each line. Column 12 contains an abbreviation code for the type of vaccine. A student's 7-digit ID number starts at column 14, and their name begins at column 22. A student is "done" with their appointments after one dose of vaccine code 'J', or after two doses of any other vaccine (the same student may appear on two different rows; if they do, their vaccine code will be the same each time).

(a) Implement the following function to extract student IDs, names, and vaccine codes. As an example, the outputs for the above file snippet would look like this:

$$\text{ids} = \begin{bmatrix} 3141592 \\ 2718282 \\ 1618034 \\ 2718282 \end{bmatrix} \quad \text{codenames} = \begin{Bmatrix} \text{'M', 'Cody Clay'} \\ \text{'P', 'Mae Goodwin'} \\ \text{'J', 'Hakim Umayya'} \\ \text{'P', 'Mae Goodwin'} \end{Bmatrix}$$

Hint: If you have trouble with the file operations, start by writing code to process a single line of text to produce a single row of output.

```
function [ids, codenames] = readVaxRecords(filename)
% Read and parse student vaccination records from file `filename`.
% The student ID from each line will be stored in the corresponding element
% of `ids` (a numeric array), while the vaccine code and student's name
% are stored in columns 1 & 2 of that row in 2D cell array `codenames`.
```

(b) To determine which students are “done” with their vaccine sequence, we need a way to determine whether a student (with code other than ‘J’) appears twice in the dataset. Observe that if records were sorted by ID rather than by date, then students who occur twice will have their ID on adjacent rows.

Write a script to print the names of students who are “done” with their vaccine sequence, based on data in a file named `vaccinations.txt`. Your script should make effective use of `readVaxRecords()` and `sort()`. Do not print a student’s name more than once.

Example: For the file on the previous page, this script should output:

```
Hakim Umayya  
Mae Goodwin
```