Name: _____ NetID: _____

(Legibly print **last name**, first name, middle name)

Statement of integrity:

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 in this examination.**

Circle your discussion section:

Section	Day and Time	Instructor
201	W 10:10 AM - 11:00 AM	Kelly Cheng
202	W 11:20 AM - 12:10 PM	Xinran Zhu
203	W 12:25 PM - 1:15 PM	Blaire Yu
204	W 1:30 PM - 2:20 PM	Blaire Yu
205	W 2:40 PM - 3:30 PM	Claire Liang
206	W 3:45 PM - 4:35 PM	Claire Liang

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.
- Raise your hand if you have any questions.
- Use the back of the pages if you need additional space.
- 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: `min`, `max`, `sum`, `rem`, `floor`, `ceil`, `rand`, `zeros`, `ones`, `linspace`, `length`, `size`, `input`, `fprintf`, `disp`, `randi`, `size`, `strcmp`, `uint8`, `double`, `fopen`, `feof`, `fgetl`, `fclose`

Examples:
`cell(3,2)` → a 3-by-2 cell array, each cell is the empty numeric vector [ ]
`rand()` → a random number between 0 and 1, not inclusive
`floor(6.9)`, `floor(6)` → 6, rounds down to the nearest integer
`ceil(8.1)`, `ceil(9)` → 9, rounds up to the nearest integer
`length([2 4 8])` → 3, length of a vector
`zeros(1,4)` → 1 row 4 columns of zeros
`[nr,nc,np]=size(M)` → dimensions of `M`: `nr` rows, `nc` columns, `np` planes or panes
`uint8(4.7)` → the integer (type uint8) value 5
`strcmp('cat', 'Cat')` → 0, the two char arrays are not identical
`randi(6)` → a random integer between 1 and 6, inclusive

Use this page for scratch work.

Question 1 (13 points)

Question 1.1 (9 points)

Circle all lines that would throw an error.

```
a = uint8(50) + 20;

b = [1, 2, 3].*[0.5, 0, -1];

c = sum(5 == [5, 6, 7]);

d = {[4,1]; [3,4,5,6], true};

e = [[4,1]; [3,4,5,6]];

f = ['4','1','3','4','5','6'];

g = ['4' {'1' ['3' '4']}];

h = {'4' {'1' '3 4'}};

i = [[4,1]; '[3 4]' ;[5, 6]];
```

Question 1.2 (4 points)

Write in the box the array X after running the following code. If an error would occur during execution, write the word "error" in the box.

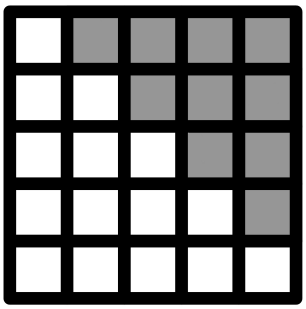
```
B = [0 2 2 2;
     1 0 2 2;
     1 1 0 2;
     1 1 1 0];
X = funFunction(B);

function A = funFunction(A)
   [nr, nc] = size(A);
   for i = 1:nr
       for j = 1:nc
           num = A(i, j);
           A(i, j) = A(j, i);
           A(j, i) = num;
       end
   end
end
```

Answer:

Use this page for scratch work.

Question 2 (12 points)

Complete the following script that takes n as an input and creates a 2D array, `A`, of size n-by-n storing integers. Each entry of `A` in the upper triangular region (gray boxes) will be a positive integer and each entry of `A` not in the upper triangular region (white boxes) will be zero.



Fill in the upper triangular region as follows:

1. Fill in each row of the upper triangular region, starting with the first row and ending with the second to last row.
2. Within each row, fill in the entries from left to right.
3. The `i`th entry you fill in should store `i`. For example, the 1st entry you fill in should store 1, the 2nd entry you fill in should store 2, ...

Example: if n = 4, then $A = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 0 & 0 & 4 & 5 \\ 0 & 0 & 0 & 6 \\ 0 & 0 & 0 & 0 \end{bmatrix}$

No built-in MATLAB functions are allowed on this problem other than `input` and `zeros`.

```
n = input('Please input an integer larger than 0: ');
```

Use this page for scratch work.

Question 3 (26 points)

Question 3.1 (6 points)

Suppose you will work with a text file that contains multiple rows of only lower case letters. As an example, the file may look like:

```
thetheoughtsinchcoco
cofiltenghercocough
fhenty
```

Write a function that stores all letters in the file into a 1D char array. Start by storing the first row of the file in a char array and append each of the following rows to this same char array. The char array that should be created using the example file listed above would be

```
st = 'thetheoughtsinchcococofiltenghercocoughfhenty'
```

Complete the function below by filling in the skeleton. In addition to filling in the blanks, you may add code above and within the loop.

```
function st = file2charArr(fileName)
% Reads all rows of a file and stores all the letters in a 1D char array
% fileName: 1D char array corresponding to the file name.
            You may assume that the file has at least one line.
% st: 1D char array storing all the letters of the input file

fid = fopen(_________________________, 'r');




while ~feof(_________________________)




end

_________________________;
```

Use this page for scratch work.

Question 3.2 (20 points)

Design a function (called `count_noShare`) that takes two inputs and has one output. The inputs are a 1D char array storing only lower case letters (call it `st`) and another 1D char array storing only lower case letters (call it `query`). The output should be a number (call it `count`) storing how many times `query` shows up in st. If `query` does not show up in `st` or `query` is longer than `st` then `count` should equal 0.

Loop through `st` from left to right and note that character sharing is not allowed. That means if `st = 'xcococoy'` and `query = 'coco'`, then `count = 1` because the 'coco' in `C(2:5)` and `C(4:7)` share characters so only the first found occurrence will be counted when you are looping from left to right.

Another example: if `st = 'xcococococoy'` and `query = 'coco'` then `count = 2`.

Write the function header (comments are not necessary) and complete the function so it follows the specifications listed above.

Use this page for scratch work.

Question 4 (22 points)

You are the substitute teacher for the day and the school sent you a 2D cell array storing a seating chart for the room. Write a function called `nameSearch` to search the given cell array for all names that contain a certain letter exactly once.

The cell array may look like:

```
seating = {'CHER',    'PRINCE',  'AALIYAH', 'ARIANA';
           '',        'GAGA',    'BJORK',   'KHALID';
           'RIHANNA', 'MADONNA', '',        'LIZZO'};
```

See the bottom of this page for an example. Write the following function:

```
function names = nameSearch(seating, ch)
% Returns a 1D cell array of all students in the seating chart who have the
% letter ch in their name exactly once. If there are no names that contain
% ch once, names should be an empty cell array.
% seating: 2D cell array where each cell stores a 1D char array of a student's
%          name. Empty seats contain empty char arrays.
% ch: the letter to search for. ch stores a single character (type char).
% names: 1D cell array of all students' names that contain ch exactly once.
% NOTE: assume that all letters in seating and ch are capital letters.
```

In the example to the right (corresponding to the cell array defined above), the call:

```
names = nameSearch(seating, 'N')
```

should create the 1D cell array `names` storing the names: `PRINCE` and `ARIANA`

Use this page for scratch work.

Question 5 (27 points)

The built-in MATLAB function `randperm` is not allowed. `randi` is allowed and may be useful in this problem.

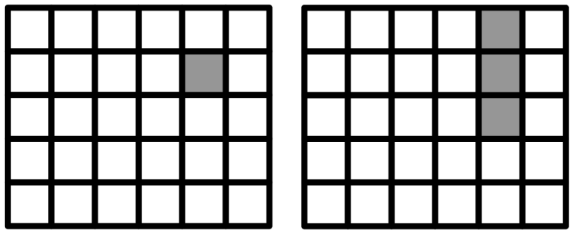
Given a 3D `uint8` array of color image data (`orig`), a 0-1 array (`key`) with the same number of row and columns as `orig`, and a numeric scalar (`nb`), create an array (`new`) that is the same as `orig` except as follows:

1. Any pixel of `new` whose corresponding element in `key` equals 1 should be replaced. At each pixel of `new` that should be replaced, each color intensity (red, green, and blue) should be replaced by the average of the corresponding color intensity in `orig` and the corresponding color intensities in the pixels of `orig` that are directly above and directly below the pixel that is being replaced, one from the top and one from the bottom (so nominally 3 intensities participate in the average). See the image below for an example. If a pixel should be replaced but it's corresponding pixel in orig does not have a neighbor above or below, use the average of the pixel and the neighbor it does have.

    - NOTE: a pixel of the image and an element of key are corresponding if they are in the same row and column in both arrays.

2. Choose `nb` distinct random columns in the image. Each pixel in each of those chosen columns will become black pixels. Recall, a black pixel is represented by [0, 0, 0].

Complete the function according to the above procedure on the next page.



Pixel in `new` that should be replaced

Pixels in `orig` that contribute to the average

Use this page for scratch work.

Question 5, continued

```
function new = modifyImg(orig, key, nb)
% Modifies an image based on the locations in key that store the value 1
% and changes nb random columns to black.
% orig: uint8 3D array corresponding to the original image.
% key: 0-1 2D array with the same number of rows and columns as orig.
% nb: number of columns to randomly turn black. Assume nb is an
%     integer greater than 0 and less than the number of columns of orig.
% new: uint8 3D array the same size as orig with modified pixels and nb
%      columns turned black.
```

Use this page for scratch work.