Name: _____

GTID# (903…): _____

CS 1371
Practice Exam 2
Form A
SOLUTIONS

-Write your name and GTID on every page or **you could lose 5 points on the exam**.
-Check the board at the front of the room for any updates/clarifications during the exam.
-You may not use the MATLAB string class
-All logical values should be denoted `true` or `false` NOT 0 or 1
-An appendix containing useful information is provided separately.
-No credit will be given for comments, but if they are useful for you, feel free to use them.
-If there is a problem that you cannot figure out, skip it and move on. Be mindful of how much each question is worth.
-If you run out of room, use the back page. Be sure to mark that you are continuing your answer on the back page!
-You have fifty minutes. Good luck!

Please fill in the entirety of the circle for multiple choice questions!!
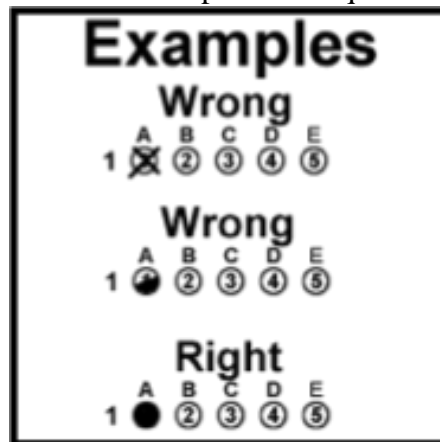
I pledge that I have neither given nor received help on this exam:

_____   _____

Signature   Date

Problem 1. Answer the following tracing questions. (25 points)

```
1     function sums = doSomething(arr)
2     [rows, cols] = size(arr);
3     sums = [];
4     for a = 1:rows
5         temp = 0;
6         for b = 1:cols
7             temp = temp + arr(b, a);
8         end
9         sums = [sums temp];
10    end
11    end
```

Assume that the above function is defined in the current directory. The following code is then run in the Command Window.

```
>> arr = [1 2 3; 4 5 6; 7 8 9];
>> sums = doSomething(arr);
```

   a. After the function is run, what is the size of `sums`? (6 points)

       Number of rows: 1

       Number of columns: 3

   b. The following lines of code are run in the command window.

```
>> newArr = [1 2 3 4; 5 6 7 8; 9 10 11 12]
>> sums = doSomething(newArr)
```

      The function will now error. On which line does the error occur? Explain. (7 points)

It will error on line 7. This will error because it will index `arr` out of bounds since the column indices are being used to index the rows.

Answer the following short coding question. Do NOT write functions and do NOT hardcode!

c. Given two vectors of doubles, `vec1` (1xN) and `vec2` (1xM), count the number of elements they have in common and store this number in `num`. There will be no repeated elements within a single vector (for example, `vec1` cannot have more than one 3). Note: the `unique()` and `contains()` functions are banned. (12 points)

<u>Test Case:</u>
```
vec1 = [2 5 1 8 4]
vec2 = [2 9 4 3]
num => 2
```

```
num = 0;

for i = 1:length(vec1)

        for j = 1:length(vec2)

                if vec1(i) == vec2(j)

                        num = num + 1;

                end

        end

end
```

Problem 2. Given the following function, answer the questions that follow. (20 points)

The following function will take in a MxN double array, M > 1, N > 1.

```
1     function out = arrayFun(in)
2     if prod(prod(in)) >= 1
3         in = 2 .* in
4         if in(1) == 1 | in(1) == 0
5             out = 1
6         elseif in(1) == 2
7             out = 2
8         else
9             out = 3
10        end
11    elseif any(min(in) > 1)
12        out = 'big NUM'
13    elseif all(mean(in) > 200)
14        out = 'best possible out'
15    end
16    out = out(1)
17    end
```

a. Assuming the function above is defined in the current directory and the following code is run in the Command Window, what are the class and value of `out2`? (5 points)

```
>> in2 = ones(5, 4)
>> out2 = arrayFun(in2)
```

Class: double

Value: 2

b. Suppose the following code is run in the command window:

```
>> out2 = arrayFun([10 6; 0 0])
```

Would the function error? If so, briefly (1-2 sentences) explain the error. If not, write the output. (5 points)

The function would error because the variable `out` is undefined when it is used on line 16. This happens because none of the conditions of the if statement are met and there is no other time where `out` is assigned a value before it is indexed on line 16.

    c. Write the necessary code to convert lines 4-10 to a `switch` statement. (10 points)

```
switch in(1)
     case {1, 0}
          out = 1;
     case 2
          out = 2;
     otherwise
          out = 3;
end
```

Problem 3: Answer the following short coding questions. <u>Do NOT write functions and do NOT hardcode!</u> (25 points)

a. Convert the following `if` statement into a `switch` statement. In the code, `pumpkins` represents the number of pumpkins you have. Remember, there can only be a positive integer (1, 2, 3…) number of pumpkins. (11 points)

```
if pumpkins > 5
     out = 'too many'
elseif pumpkins < 5
     out = 'not enough'
else
     out = 'just right'
end
```

<u>Your conversion:</u>

```
switch pumpkins
      case {1, 2, 3, 4}
            out = 'not enough';
      case 5
            out = 'just right';
      otherwise
            out = 'too many';
   end
```

b. You are given two vectors, `vec1` and `vec2`. If the vectors' lengths are equal, check if the vectors are identical. If so, output either vector; if they aren't identical, output how many pairs of corresponding elements were different. If the vectors have different lengths, output the longer vector. Assign the output to `out`. Hint: a nested conditional may be helpful and you do not need iteration. (14 points)

Test Case:
```
vec1 = [3 4 5]
vec2 = [2 4 8]
out => 2
```

```
if length(vec1) == length(vec2)
    if all(vec1 == vec2)
        out = vec1;
    else
        out = length(sum(vec1 ~= vec2));
    end
elseif length(vec1) > length(vec2)
    out = vec1;
else
    out = vec2;
end
```

Problem 4. Write the following function. Your function should work for all possible inputs as specified by the problem. Do NOT hard code using any of the examples. (30 points)

Function Name: `partyMonster`

Inputs(2) : (double) Nx4 array of information about different party guests

(char) Character vector of the form 'Sam wants # dollars per guest'

Outputs(2): (double) Updated Nx4 array

(logical) Whether all of the guests brought the dollar amount Sam requested

Function Description:

It's Sam's Birthday, and you've decided to throw a surprise birthday party for him. You are given an array where each row represents a separate person and what they are bringing. The first column will be the amount of money they are giving as a **gift**, the second column will be the number of **cookies**, the third column will be the number of **cupcakes**, and the last column will be the number of **brownies**.
1. Find the negative values in the array and make them positive.
2. Rearrange the rows according to the number of **brownies** in descending order. The row with the highest number of brownies should be on top, and the lowest on the bottom.
3. Find the maximum of each column and append these values to the bottom of the array.
4. If every guest brings more money than Sam asked for as a gift, output a logical true for the second output. Otherwise, output a logical false.

Test Case:

```
>> arr = [30,  10, -10,  5;
          -16, -20,  30,  0;
           50,   5,   5, 10;
           20, -20, -25, 20]
>> str = Sam wants 25 dollars per guest'
>> [newArr, check] = partyMonster(arr, str)
newArr => [20, 20, 25, 20;
           50,  5,  5, 10;
           30, 10, 10,  5;
           16, 20, 30,  0;
           50, 20, 30, 20]
check => false
```

Name: _____

GTID# (903…): _____

Write your code for Problem 4 here.

```matlab
function [newArr, check] = partyMonster(arr, str)
arr(arr < -1) = arr(arr < -1) .* -1;
[~, inds] = sort(arr(:,4),'descend');
newArr = arr(inds, :);
maxVals = max(newArr);
newArr = [newArr; maxVals];
[~, rest] = strtok(str);
[~, rest] = strtok(rest);
num = strtok(rest);
num = str2num(num);
if all(newArr(:,1) > num)
    check = true;
else
    check = false;
end
end
```

Name: _____

GTID# (903…): _____

EXTRA SPACE: if you use this space, be sure to label which question you are answering. In the original question space, CLEARLY leave a note that you've continued the answer here.