Name: _____

GTID (ex: 903XXXXXX): _____

CS 1371
Practice Exam 2
Form B
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.
- All logical values should be denoted using `true` or `false` NOT 1 or 0
- An appendix containing useful information is provided separately.
- If there is a problem that you cannot figure out, skip it and move on. Be mindful of how many points 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!

I pledge that I have neither given nor received help on this exam:

_____ _____

Signature Date

Problem 1. Answer the following tracing questions. (20 points)

```
1  function [newArr, maxSum] = bestSum(arr)
2  [r,c] = size(arr);
3  newArr = [];
4  for i = 1:c
5        temp = 0;
6        for j = 1:r
7              temp = temp + arr(j, i);
8        end
9        newArr = [newArr temp];
10       % i = i + 1;
11 end
12
13 ind = 2;
14 maxSum = newArr(1);
15
16 while ind <= length(newArr)
17       if newArr(ind) > maxSum
18             maxSum = newArr(ind);
19       end
20       ind = ind + 1;
21 end
22 end
```

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

```
>> arr = [2 0; 3 1; 2 5; 4 8];
>> [new1, max1] = bestSum(arr);
```

a. After the function is run, what will be the values of `new1` and `max1`? (3 points each)

```
new1 = [11 14]
```

```
max1 = 14
```

b. Using the same input given in part a, how many **total** times will the for-loop on lines 6-8 run? (3 points)

It will iterate through 8 total times.

Name: _____

GTID (ex: 903XXXXXX): _____

c. Suppose that Line 10 is uncommented. Does the for-loop now operate differently? Explain why or why not. If it operates differently, indicate how the values of `temp` and `newArr` change. If the code errors, indicate what the error would be. (4 points)

The for loop operate in the exact same way, and the output does not change. Since i is reset at the beginning of each iteration, and it is never used after it is altered by line 10, this does not affect the code in any way.

d. Convert the following for loop into a while loop. Assume that `vec` is a 1xN double vector. (7 points)

```
1  diffVec = [];
2  for x = 1:length(vec)-1
3   diffVec = [diffVec (vec(x+1) - vec(x))];
4  end
```

Write your code here:

```
diffVec = [];
x = 1;
while x <= length(vec) - 1
   diffVec = [diffVec (vec(x+1) - vec(x))];
   x = x + 1;
end
```

Problem 2. Answer the following multiple choice questions. (20 points)

a. Consider the following block of code. You may assume that `vec` is a 1xN double vector that exists in the scope of your current script.

```
1 mask = vec >= 0;
2 if sum(mask) == length(vec)
3 out = 'Woo';
4 elseif isequal(sum(vec(mask)), sum(vec(~mask)))
5   out = 'Halfsies';
6 else
7 out = 'No balance'
8 end
```

Choose **ALL OF THE FOLLOWING** blocks of code which accomplish the same task as the above code. (8 points)

□ Option A

```
1  mask = vec >= 0;
2  switch all(mask)
3     case {true}
4          out = 'Woo';
5     otherwise
6          switch sum(vec(mask))
7               case sum(vec(~mask))
8                    out = 'Halfsies';
9               otherwise
10                   out = 'No balance';
11         end
12 end
```

□ Option B

```
1  new = abs(vec)
2  switch vec
3     case new
4          out = 'Woo';
5     otherwise
6          mask1 = sum(vec >= 0)
7          mask2 = sum(vec < 0)
8          if mask1 == mask2
9               out = 'Halfsies';
10         else
11              out = 'No balance'
12         end
13 end
```

□ Option C

```
1 mask = vec >= 0;
2 switch vec
3     case sum(mask)== length(vec)
4          out = 'Woo';
5     case isequal(sum(vec(mask)), sum(vec(~mask)))
6          out = 'Halfsies';
7     otherwise
8          out = 'No balance';
9 end
```

Name: _____

GTID (ex: 903XXXXXX): _____

b. You are given two vectors: a 1xN char vector called `teams` with the name of two teams **separated by a single space**, and a 1x2 double vector called `wins` with the respective number of wins that each team has. You want to figure out which team to bet on, so write a few lines of code in accordance with the following rules to help you:

- If the first team has less than 10 wins, you immediately bet on the second one.
- If the second team has more wins than the first team you bet on the second team.
- If neither of the above two conditions are true, bet on the first team.

Store the **name of the team** you want to bet on in a variable named `bet`. (12 points)

Test Case:

```
teams = 'Chelsea Liverpool'
wins = [4 8]

>> bet = 'Liverpool'
```

Write your code here:

```
[team1, team2] = strtok(teams);
team2 = strtok(team2);
if wins(1) < 10
      bet = team2;
elseif wins(2) > wins(1)
      bet = team2;
else
      bet = team1;
end
```

Problem 3. Answer the following tracing questions. (30 points)

```
1  function [forecast, daysOff] = snowDay(city, precip, temp)
2  if temp <= 32 & precip > 1
3     forecast = sprintf('%d inches of snow!', precip);
4     switch city
5           case {'Atlanta', 'Savannah', 'Tampa', 'Orlando'}
6                 daysOff = 10
7           case {'Boston', 'New York', 'Ithaca'}
8                 daysOff = 1;
9           otherwise
10                daysOff = 3;
11    end
12 elseif temp <= 32
13    forecast = sprintf('%d degrees. Bundle up!', temp);
14    daysOff = 0;
15 else
16    forecast = sprintf('%d degrees.', temp);
17    daysOff = 0;
18 end
19 end
```

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

```
>> [forecast1, daysOff1] = snowDay('Atlanta', 0, 30);
```

a. What are the values of forecast1 and daysOff1? (10 points)

```
forecast1 = '30 degrees. Bundle up!'
```

```
daysOff1 = 0
```

Name: _____

GTID (ex: 903XXXXXX): _____

b. Please give a set of input values for the above function that could produce the following output values. (12 points)

```
>> [forecast2, daysOff2] = snowDay(city, precip, temp);
forecast2 => '5 inches of snow!'
daysOff2 => 3
```

city = any city that is not one of the seven in the switch statement (Any answer that is **NOT** 'Atlanta', 'Savannah', 'Tampa', 'Orlando', 'Boston', 'New York', 'Ithaca')

precip = 5

temp = any temperature that is **less than or equal to 32**

c. Lines 15 – 17 are deleted and the following line of code is run in the command window.

```
>> [forecast3, daysOff3] = snowDay('New York', 2, 50);
```

After the lines are deleted, will the code error? Please select the correct choice and answer the follow-up question. (8 points)

◯ The code will error

Explanation: Output arguments are not assigned

◯ The code will not error

```
forecast3 =
daysOff3 =
```

Problem 4. Write the following function. <u>Do NOT hardcode!</u> (30 points)

Function Name: `matrixManipulator`

Inputs (2): (double) NxM array
(double) 1xM vector

Outputs (2): (double) a manipulated array
(double) a calculated value

Function Description: You are to manipulate a given NxM array and a 1xM vector.

<u>Write a function that completes the following steps in order:</u>

- Sort the columns of `arr` in ascending order according to the values in the first row.
- Replace all negative values in `arr` with the equivalent positive value (i.e. -1 should become 1, -7 should become 7, etc.)
- Delete all rows that have a maximum value less than 4.
- Replace every even row in `arr` with `vec`.
  Note: `arr` and `vec` will have the same number of columns.
- Find the sum of the right half of `arr` **excluding** the first row and make the resulting value the second output of your function.
  Note: `arr` is guaranteed to have an even number of columns.

Test Case:

```
arr =         [3     2     1     4;
              -5    -1     7     0;
             -11    -5     0     9;
               4     4     4     2;
               0     1     3     0]

vec = [20, 10, 20, 10]

>>  [manipulatedArr, num] = matrixManipulator(arr, vec)


manipulatedArr => [1     2     3     4;
                   20    10    20    10;
                    0     5    11     9;
                   20    10    20    10]

num=> 80
```

Name: _____

GTID (ex: 903XXXXXX): _____

Write your FUNCTION here:

```
function [manipulatedArr, num] = matrixManipulator(arr, vec)
[~, inds] = sort(arr(1, :));
arr = arr(:, inds);
arr(arr < 0) = arr(arr < 0).*-1;
maxVals = max(arr,[],2);
mask = maxVals < 4;
arr(mask, :) = [];
[r,~] = size(arr);
for i = 2:2:r
    arr(i,:) = vec;
end
manipulatedArr = arr;
num = sum(sum(arr(2:end,end/2+1:end)));
end
```

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 have continued the answer here.