Q1:

Problem 1. Answer the following conceptual questions. (20 Points)

a. Determine whether the following function headers are valid. For EACH header circle whether they are *valid* or *invalid.* If the function header is *invalid,* rewrite the CORRECTED function header using the same function and variable names. (2 points each)

```
Function avocado(in1, in2)
```

Valid Invalid ____function avocado(in1, in2)_____

```
function [] = fluffers()
```

Valid Invalid _____

```
function [out1, out2] = doggos(dog1 dog2)
```

Valid Invalid ____function [out1, out2] = doggos(dog1, dog2)_____

b. Consider the following function that takes in a 1XN vector of doubles and answer the questions below.

```
1    function average = findAverage(vec)
2    sumOfVec = sum(vec);
3    average = sumOfVec./5
4    end
```

i. Will the function above find the average of the vector for every vector input? Explain your answer (3 points).

No, because not all vectors are of length 5 and this only accounts for vectors of length 5. This is an example of hardcoding.

c. Consider the following function and answer the questions below.

```
1     function [cost,time] = phonePlan(calls, texts)
2     temp = calls + texts;
3     cost = temp .* 2;
4     time = calculate(calls);
5     cost = cost + time;
6     end
7
8     function [out] = calculate(calls)
9     time = calls .* 3;
10    out = time + 1;
11    end
```

The following code is run in the Command Window:

```
>> textsSent = 2;
>> [wasted, cost] = phonePlan(3,textsSent);
```

i. Complete the table below with the values of each variable in the listed workspaces. If the variable does not exist in the workspace then write DNE. (2 points each)

	Command Window	phonePlan	calculate
`cost`	10	20	DNE
`time`	DNE	10	9

ii. If line 5 in the code above is changed to `cost + time = cost`, will the code still run as expected (3 points)?

a. Yes, the code will run as expected.
b. No, the code will error.
c. No, the value of `time` will be different.
d. No, the value of `cost` will be different.

Q2:

a. Class: `'char'`
Value: `'Whos thE GoAt BUzz'`

b. Yes, the function would error on line 4 because the input `word2` is a length 6 and the `pos:select` vector will be a length of 8 causing a dimension mismatch.

c. `'Whos the Goat Buzz'`

d. The code would error on line 8 because the variable `letters` will result in a mask that includes the punctuation and after deleting the puncution the `letters` mask and `str` won't be the same length causing an index out of bounds error (dimension mismatch error also accepted).

Q3:

a.

```
totalVec = zeros(1, length(vec1).*3);
totalVec(1:3:end) = vec1;
totalVec(2:3:end) = vec2;
totalVec(3:3:end) = vec3;
[ ~ , smallest] = min(totalVec);
```

b.

```
mask = mod(vec, 2) == 0;
evenCount = sum(mask);
vec(1:end/2) = vec(1:end/2) - evenCount;
vec(end/2+1:end) = vec(end:-1:end/2+1);
newVec = vec;
```

c.

The code will NOT error.

```
num => 4
```

Q4:

```
function [newArr, check] = partyMonster(arr, str)
mask = arr < 0;
arr(mask) = arr(mask) .* -1;
[~,ind] = sort(arr(:,end),'descend');
newArr = arr(ind, :);
newArr = [newArr ; max(newArr)];
[~, rest] = strtok(str);
[~, rest] = strtok(rest);
num = strtok(rest);
mask = newArr(:,1) > str2num(num);
check = all(mask)
end
```

Other ways to find number:

-Mask to find spaces, index between 1 after second space and 1 before third space

-Mask between ASCII values 48 and 57, apply to string to get out number