

Your Name: Answers

Checkpoint Quiz 7 (with make up questions for Quiz 1,5,6):

1.2 Suppose Q is a vector. Write any correct matlab code to change all the 0's in Q to be 2's. That is, if Q starts off as [0 1 4 1 0], then it should become [2 1 4 1 2]. Your code should work with *any* vector Q, not just the example values above.

$$Q(Q == 0) = 2$$

5. Define a function foo that takes an array of sample values of a function as returns the "local-min" locations of the function; positions where the function is greater than the previous or next value. You may assume that all elements of the input are positive, and that no two elements of the input are equal: For example

<pre>>>foo([1 3 5 4 2]) [1,5]</pre> <p>Because the first and last numbers are smaller than their neighbors</p>	<pre>>>foo([4 3 -1 1]) [3]</pre> <p>Because the thid number is smaller than the second or fourth</p>	<pre>>>foo([25 5 14 8 16]) [2 4]</pre> <p>Because the second and fourth numbers are local minima.</p>
--	--	---

I will get you started:

function localMinima = foo(A)

```
k = []
for ix = 2: length(A) - 1
    if A(ix) < A(ix-1) & A(ix) < A(ix+1)
        k = [k ix]
    end
end
```

```
if (A(end-1) > A(end))
    k = [k length(A)]
end
```

OR
~~B = [A(1) A A(end)]~~
~~find(B(2:end-1) > B(3:end) & B(2:end-1) > B(1:end-2))~~
~~B(x:end-1) > B(1:end)~~

6. Define a function bar that takes the name of a function and a value x and computes the difference of the value of the function at x-1 and at x. Examples of what your program should produce:

<pre>>>bar('sqrt',2) 0.414</pre> <p>Because sqrt(2)-sqrt(1) = 0.414.</p>	<pre>>>bar('abs',4) 1</pre> <p>Because 4 - 3 = 1.</p>
--	---

ALTERNATE ANS FOR #5
 B = [A(1) A A(end+1)]
 mins = B(2:end-1) > B(3:end) & B(2:end-1) > B(1:end-2)
 localMinima = find(mins)

I will get you started:

function diff = bar(funcName, x)

```
diff = feval(funcName, x) - feval(funcName, x-1)
```

7. Define a function monotonic that takes the name of a function, and a list of x-values (in increasing order), and returns 1 if the function value is always increasing over that range of x-values, and 0 otherwise. You must correctly define the function header in addition to the function. Your function should return, for example:

<pre>>>monotonic('sqrt',[0.1 0.2 0.3 0.4 0.5]) 1</pre> <p>Because the sqrt of these values always increases</p>	<pre>>>monotonic('cos',[-0.2 -0.1 0 0.1 0.2]) 0</pre> <p>Because the cos function rises then falls over this range..</p>
---	--

function m = monotonic(funcName, x)

```
m = 1
for ix = 1: length(x) - 1
    if feval(funcName, x(ix+1)) < feval(funcName, x(ix))
        m = 0
    end
end
```