Algorithm Correctness Argument Example

function sortedVctrRS = exchangNumericSortFN(vctrToBeSortedPR)

% exchangNumericSortFN sorts a numeric vector in ascending order using an

% exchange (or bubble) sort.

%

% SYNTAX

% If vctrToBeSortedPR is a numeric vector, sortedVctrRS has the same values

% as vctrToBeSortedPR except they are now in order by numeric value. The sort

% is stable, i.e. vector elements with exactly the same value retain the

% order in which they appear in vctrToBeSortedPR.

% REQUIREMENTS

% R01 If vctrToBeSortedPR IS NOT a numeric vector exchangNumericSortFN displays:

%

% ABORTING exchangNumericSortFN:

% Input vctrToBeSortedPR is not a numeric vector.

%

% and returns to the Command Window prompt.

%

% R02 If vctrToBeSortedPR IS a numeric vector exchangNumericSortFN returns

% sortedVctrRS to the caller. The values in sortedVctrRS are

% the same as those in vctrToBeSortedPR but they are now in

% non-decsending order.

%

% R03 The order of equal values in vctrToBeSortedPR has not been distrubed.

%

% R04 The sorting of vctrToBeSortedPR is accomplished by exchanging

% seccessive values for which the first value is larger than the

% second.

%

% Sample invocations:

% ------------------

% (1) vctrToBeSortedPR: 33

% sortedVctrRS: 33

%

% (2) vctrToBeSortedPR: [1 3 2 4]

% sortedVctrRS: [1 2 3 4]

%

% (3) vctrToBeSortedPR: [-1.1 -2.2 -3.3 -4.4 -5.5 -6.6]

% sortedVctrRS: [-6.6 -5.5 -4.4 -3.3 -2.2 -1.1]

%

% (4) vctrToBeSortedPR: [1 2; 3 4]

% ABORTING exchangNumericSortFN:

% Input vctrToBeSortedPR is not a numeric vector.

%

% HISTORY

% Date Author Description

% ---- ------ -----------

% 09/18/15 Frank Ackerman Initial version

% 12/03/15 Frank Ackerman Name changes

% SIGNIFICANT DATA

% vctrBeingSorted - set initiallly to vctrToBeSortedPR

% didAnExchangeFlg - initially set 'true' but then is 'false' only when a

% complete pass across vctrBeingSorted does not result

% in any exchanges.

%

% SOLUTION ANALYSIS

% Make a series of passes through vctrBeingSorted and compare the values of

% successive pairs. If the first value is greater than the second,

% exchange the values. If a pass does not make any exchanges, the values

% of vctrBeingSorted are in non-descending order.

%

% ALGORITHM

% A01 If (vctrToBeSortedPR is not a numeric vector)

% A02 Call error(with aborting message R01)

% End

%

% A03 Set vctrBeingSorted to vctrToBeSortedPR;

% Set didAnExchangeFlg to true;

%

% A04 While (didAnExchangeFlg)

%

% A05 Set ndx to 1;

% Set didAnExchangeFlg to false;

% A06 While (true)

%

% A07 If ( ndx is equal to length(vctrBeingSorted) )

% Break

% End %If pass across vctrBeingSorted is completed

%

% A08 If (vctrBeingSorted(ndx) greater than

% vctrBeingSorted(ndx+1))

% Exchange values of vctrBeingSorted(ndx) and

% vctrBeingSorted(ndx+1);

% Set didAnExchangeFlg to true;

% End %If

% A09 Increment ndx;

% End %While making a single pass

% A10 Continue; %to possibly make another pass

%

% End %While making passes

%

% A11 Set sortedVctrRS to vctrBeingSorted;

% Return;

%

%

%

%

%

% ALGORITHM CORRECTNESS ARGUMENT (show algorithm implements each

% requirement)

% CR01 By algorithm block A01.

% CR02 Consider 2 cases:

% (a) vctrBeingSorted is a numeric vector whose values are already

% in non-descending order.

% (b) the values in vctrBeingSorted are not in non-descending order

% For both cases all pairs of values of vctrBeingSorted are

% considered in algorithm blocks A04 through A10.

% For case(a):

% (1) didAnExchangeFlg is set to false in A05. The only place it

% is changed is in A08, but it will only be changed if there

% is a pair of values that is not in order, but the

% assumption here is that they are in order.

% (2) Hence, when A10 is executed the condition at A04 will be

% false and the next block executed is A11.

% For case(b):

% (1) By allgorithm blocks A06, A09, and A10 vctrBeingSorted will

% be scanned until any adjacent pairs of values in which the

% first is greater than the second will be exchanged. When

% all the values in vctrBeingSorted are in non-descending

% order this cannot occur.

% CR03 The order of values is altered only in algorithm block A08 and only

% if the values are not equal.

% CR04 By algorithm block A08.

% ---------------------------------------------------------------------

 if ( isNotAnumericVectrFN(vctrToBeSortedPR) ) %L01

 error(['ABORTING exchangNumericSortFN: ', ... %L02

 'Input vctrToBeSortedPR is not a numeric vector.']);

 end %if

 vctrBeingSorted = vctrToBeSortedPR; %L03

 didAnExchangeFlg = true;

 while (didAnExchangeFlg) %L04

 ndx = 1; %L05

 didAnExchangeFlg = false;

 while (true) %L06

 if ( ndx == length(vctrBeingSorted) ) %L07

 break;

 end %if pass is complete

 if ( vctrBeingSorted(ndx) > vctrBeingSorted(ndx+1) ) %L08

 tempVar = vctrBeingSorted(ndx);

 vctrBeingSorted(ndx) = vctrBeingSorted(ndx+1);

 vctrBeingSorted(ndx+1) = tempVar;

 didAnExchangeFlg = true;

 end %if an exchange should occur

 ndx = ndx + 1; %L09

 end %while making a single pass

 continue; %to possibly make another pass %L10

 end %while making passes

 sortedVctrRS = vctrBeingSorted; %L11

 return;

end %function exchangNumericSortFN()

%end exchangNumericSortFN.m