005-Topic: Array Manipulation

Course Learning Objectives: Being able to use programming structures and MATLAB high-level tools to manipulate 1D, 2D, and multidimensional arrays[1]

1.- Transfer all values of 1D array A to 1D array B

```
% Given
A=[10, 9, 8, 7, 6, 5, 4, 3, 2, 1]   % transfer  values to ➔ B
```

(a) By loops

(b) By array operations (vectorization)

ANSWER

```
%Given
A=[10,9,8,7,6,5,4,3,2,1];

(a)  for j=1:1:10
         B(j)=A(j);
     end

(b)  B=A;               % This a Ferrari statement
```

2.- Transfer all values of 2D array A to 2D array B

```
A=[10, 9, 8,
    7, 6, 5,
   4, 3, 2]  ➔➔➔ B
```

ANSWER

```
(c)  for j=1:1:3
             for i=1:1:3
                 B(i,j)=A(i,j)
             end
```

[1] Course Learning Outcomes (CO):
CO-2: Use algorithm development tools to develop a computer solution for engineering related problems, which include the skills necessary to:
(c) Apply the decision and repetition structures to the solution of problems.
Student Outcomes:
(a) an ability to apply knowledge of mathematics, science, and engineering
(e) an ability to identify, formulate, and solve engineering problems
(k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.

```
end
```

```
(d)  B=A       % This is a Ferrari statement
```

3.- Transfer selected values of array A to array B

A=[10, 9, 8, 7, 6, 5, 4, 3, 2, 1] ➔ B , where array B contains only four values

(a) By loops
(b) By array operations (vectorization)

ANSWER for (a). Please note the indexes of B are different than A. Therefore the need for two indices variables.

```
ii=1;
(a)  for jj=4:1:7
         B(ii)=A(jj)
         ii=ii+1;
     end                           % B(1), B(2),…., B(10) where are they?
```

ANSWER for (b)

1. B(1:1:4)=A(4:1:7)
2. More simple: B=A(4:1:7)   % B(1)=A(4), B(2)=A(5), … etc.

Note: If you proposed a solution for problem (a) as:

ANSWER for (a)

```
(b)  for j=4:1:7
         B(j)=A(j)
     end                           % B(1), B(2),…., B(10) where are they?
```

You failed because what is the value of B(1), B(2), B(3). If you start with B(4) and ends with B(7) MATLAB automatically assumes zero for B(1), B(2) and B(3). Similar problems has the following solution:

```
B(4:1:7)=A(4:1:7)          % B(4)=A(4), B(5)=A(5),… but the remaining B(ii) are zero
```

4.- Transfer yellow values of array A to array B

```
A=[10, 9, 8,
   7, 6, 5,
   4, 3, 2]  ➔➔➔ B              %  B is a 1D array with 3 elements
```

ANSWER

```
% Given
A=[10,9,8;7,6,5;4,3,2];
(c)  i=2; m=1             % m starts row index of B in one
     for j=1:1:3
                  B(m,j)=A(i,j)
     end

(d)  B=A(2,:)
```

5.- Transfer yellow values of array A to array B

```
          A=[10, 9, 8,
               7, 6, 5,
              4, 3, 2]  ➔➔➔ B               % B is a 2D array with only 4 elements

     ANSWER
     m=1;      % m starts row index for B in one
(e)  for i=2:1:3
        for j=1:1:2
                  B(m,j)=A(i,j)
        end
        m=m+1;
     end

(f)  B=A(2:1:3,1:1:2)            % Also B=A(2:3, 1:2)  step default is 1
```

6.- Initialize the following Matrix:

$$A = \begin{bmatrix} F & M & M & M & M \\ M & F & M & M & M \\ M & M & F & M & M \\ M & M & M & F & M \\ M & M & M & M & F \end{bmatrix}$$

```
%doc
clc, clear

% Method-1 (row-by-row)  [ this a 'cave man' method, works for small arrays

A=['FMMMM';'MFMMM';'MMFMM';'MMMFM';'MMMMF'];
```

```
% Method-2:  This is a huge better solution because it allows to handle large matrix and reflects you know the programming structures

for ii=1:1:5
   for jj=1:1:5
      if ii==jj
         A(ii,jj)='F';
      else
         A(ii,jj)='M';
      end
   end
 end

% Method-3

A(1:1:5;1:1:5)=’M’;        % All are initialized with ‘M’
for ii=1:1:5
          A(ii,ii)=’F’;        % Diagonal elements are replaced by ‘Fs’
end



% Method-4, With one index?  NO WAY JOSE
% Looks beautiful but doesn’t work at all:
for k =1:1:25
   if mod(k,6)==1
          A(k)='F';                    % ‘F’ elements are in indices: 1, 7, 13, 19, and 25
   else
          A(k)='M';
   end
end

% Why doesn’t work?
% We can reference the elements of a 2-D array with one index
% however we can’t initialize a 2-D array with only one index.
```

ONE-INDEX MANIPULATION

The following exercises take advantage of the way array elements are stored in MATLAB

7.- Consider the following array

Original

$$A = \begin{bmatrix} M & M & M & M & M \\ M & M & M & M & M \\ M & M & M & M & M \\ M & M & M & M & M \\ M & M & M & M & M \end{bmatrix}$$

Produce

$$A = \begin{bmatrix} F & M & M & M & M \\ M & F & M & M & M \\ M & M & F & M & M \\ M & M & M & F & M \\ M & M & M & M & F \end{bmatrix}$$

Given

```
% doc
clc, clear

A(1:1:5,1:1:5)='M';               % All are initialized with ‘M’ and the shape of the
                                  % array is set
```

Use one-index to replace the diagonal values with ‘F’

ANSWER-1

```
A(1:6:25)='F';
```

ANSWER-2

```
for ii=1:6:25
        A(ii)=’F’;
end
```

Note the position of the ‘F’ when one-index is used to reference each element. For this case the indices are:

1	6	11	16	21
2	7	12	17	22
3	8	13	18	23
4	9	14	19	24
5	10	15	20	25

NOTE: Once the array is initialized, with both its values and its dimensions (1D, 2D, etc.) then you can replace some values on it.

8.- Consider the following array

Original: Modified:

$$A = \begin{bmatrix} M & M & M & M & M \\ M & M & M & M & M \\ M & M & M & M & M \\ M & M & M & M & M \\ M & M & M & M & M \end{bmatrix} \qquad A = \begin{bmatrix} M & M & M & M & F \\ M & M & M & F & M \\ M & M & F & M & M \\ M & F & M & M & M \\ F & M & M & M & M \end{bmatrix}$$

Given

```
% doc
clc, clear

A(1:1:5,1:1:5)='M';        % All are initialized with ‘M’ and the shape of the
                           % array is set
```

Use one-index to replace the diagonal values with ‘F’

ANSWER-1

```
        A(5:4:21)='F';              % Stop at 21, do not go up to 25
```

ANSWER-2

```
        for ii=5:4:21
                A(ii)=’F’;
        end
```

Note the position of the ‘F’ values when one-index is used to reference each target element:

1	6	11	16	21
2	7	12	17	22
3	8	13	18	23
4	9	14	19	24
5	10	15	20	25

9.- Consider the following array (this was a question in an old final exam)

Original

$$A = \begin{bmatrix} M & M & M & M & M \\ M & M & M & M & M \\ M & M & M & M & M \\ M & M & M & M & M \\ M & M & M & M & M \end{bmatrix}$$

Produce

$$A = \begin{bmatrix} M & M & M & M & F \\ M & M & M & M & F \\ M & M & M & M & F \\ M & M & M & M & F \\ M & M & M & M & F \end{bmatrix}$$

Given

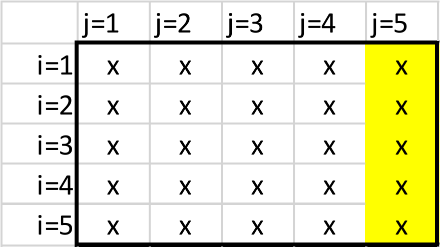
```
% doc
clc, clear

A(1:1:5,1:1:5)='M';      % All are initialized with 'M' and the shape of the
                         % array is set
```

Use

(a) Two indices to replace the values in the last column with 'F'
(b) One-index to replace the values in the last column with 'F'

ANSWER for (a)

Note the position of the 'Fs' as we reference them with two-indices. Substitution takes place only at the last column (j=5)



	j=1	j=2	j=3	j=4	j=5
i=1	x	x	x	x	x
i=2	x	x	x	x	x
i=3	x	x	x	x	x
i=4	x	x	x	x	x
i=5	x	x	x	x	x

Method-1:

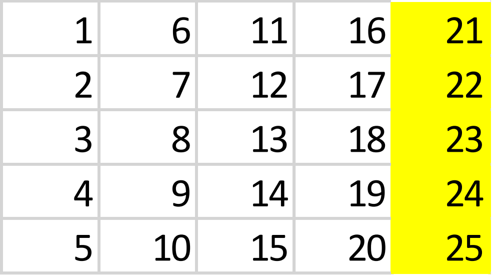
```
	jj=5
	for ii=1:1:5
		A(ii,jj)=’F’;
	end
```

Method-2:

```
	A(1:1:5,5)=’F’
```

ANSWER-for (b)

Note the position of the ‘F’ when one-index is used to reference each element:



1	6	11	16	21
2	7	12	17	22
3	8	13	18	23
4	9	14	19	24
5	10	15	20	25

Method-1:

```
	A(21:1:25)='F';
```

Method-2

```
	for ii=21:1:25
		A(ii)=’F’;
	end
```

10.- Consider the following 100x100 array (question on a final exam). Write code to modify the original array:

Original Modify to:

$$A = \begin{bmatrix} M & M & M & \dots & M \\ M & M & M & \cdots & M \\ M & M & M & \vdots & M \\ \cdots & \cdots & \cdots & & M \\ M & M & M & \cdots & M \end{bmatrix} \qquad A = \begin{bmatrix} M & M & M & \dots & F \\ M & M & M & \cdots & F \\ M & M & M & \vdots & F \\ \cdots & \cdots & \cdots & & \cdots \\ M & M & M & \cdots & F \end{bmatrix}$$

11.- Consider the following 5x5 array. Write code to modify the original array:

Original Modify to:

$$A = \begin{bmatrix} F & F & F & F & F \\ M & M & M & M & M \\ M & M & M & M & M \\ M & M & M & M & M \\ M & M & M & M & M \end{bmatrix} \qquad A = \begin{bmatrix} F & M & M & M & M \\ F & M & M & M & M \\ F & M & M & M & M \\ F & M & M & M & M \\ F & M & M & M & M \end{bmatrix}$$

Initialize the original array:

```
for ii=1:1:5
        for jj=1:1:5
              if ii==1
                     A(ii,jj)=’F’;
              else
                     A(ii,jj)=’M’;
              end
        end
end
```

Method-1:

```
% Array Initialization

clc,clear
```

```
for ii=1:1:5
    for jj=1:1:5
     if ii==1
        A(ii,jj)='F';
     else
        A(ii,jj)='M';
     end
        T(ii,jj)=A(ii,jj);  % T(ii,jj) is a temporal array
    end
end

% Array Transfer

for ii=1:1:5
    for jj=1:1:5
     A(jj,ii)=T(ii,jj);
    end
end
```

Method-2 (Ferrari approach):

```
clc,clear

% Array Initialization {Vectorization):
    jj=1:1:5;
    ii=1;
        A(ii,jj)='F';
    ii=2:1:5;
        A(ii,jj)='M';

% Array Transfer (Vectorization)
A=A';
```

SECTION 2 PROBLEMS

ARRAY MANIPULATION-Exercises

Problem 1.

Given
A=[1 2 3];
B=[3 2 1];
C=A+B;
	What is the value of C?
	ANSWER:
	C=
	4 4 4

Problem 2.

Given: >> A=[1 2 3; 2 4 6] A = 1 2 3 2 4 6 >> B=[-5 6 10; 2 0 9] B = -5 6 10 2 0 9	>> A+B % What is the output? ans = -4 8 13 4 4 15

Problem 3

Given: >> A=[1,3,5;2,4,6] A = 1 2 3 2 4 6 >> B=[2,3,4;-1,-2,-3] B = 2 3 4 -1 -2 -3	>> A.*B % What is the output? ans = 2 6 12 -2 -8 -18

Problem 4

Given : >> A=[2,4,10] A = 2 4 6 >> B=[2,2,2] B = 2 2 2	>> A./B % What is the output ? ans = 1 2 5 % num./den Right division

Problem 5

```
>> A=[2,4,6]
A =
    2    4    5
>> B=[2,2,2]
B =
    2    2    2
```

```
>> B.^A   % What is the ouput ?
ans =
    4   16   32
```

Problem 6

```
Given:
>> a=[2 4 6];
```

```
>> a*2   % What is the output ?
ans =
    4    8    12
```

Problem 7

```
Given:
>> a=[2 4 6];
```

```
>> a .* 2   % What is the output ?
ans =
    4    8    12
```

Problem 8

```
Given:
>> B=[2,4,6]
B =
    2    4    6
>> A=[3]
A =
    3
```

```
>> A.*B    % What is the output?
ans =
   6   12   18
```

Problem 9

```
>> a=[4 3 2]       % a contains 3 elements
a =
    2    3    4
>> a(4)=1                     % now a contains 4
elements
```

```
% What are the a element values?
a =
    4   3    2   1
```

Problem 10

```
>> a=[1, 2, 3]
```

```
% What are the element values of a?
```

a = 1 2 3 >> a=[a, 4, 5]	a = 1 2 3 4 5

Problem 11

>>A=[3,5,7, 8] A = 3 5 7 8 >> A(3)=[ ]	% What are the element values of a? A = 3 5 8

Problem 12

>> B=[1 3 5; 2 4 6] B = 1 3 5 2 4 6 >> B(1,:)=[ ]	% What are the element values of B? B = 2 4 6

Problem 13

>> A=[1 3 5; 2 4 6; 3 5 7] A = 1 3 5 2 4 6 3 5 7 >> A(2,3)*3	% What value is in the row 2, column 3 of A? ans = 18

Problem 14

>> A=[1 3 5; 2 4 6; 3 5 7] A = 1 3 5 2 4 6 3 5 7	>> A(4)*3 % What is the output? Ans = 9

Problem 15

>> A=[1 3 5; 2 4 6; 3 5 7] A = 1 3 5 2 4 6 3 5 7	>> A(1)*A(4)*A(7) % What is the output? Ans 15

Problem 16

>> A=[1 3 5; 2 4 6; 3 5 7] A = 1 3 5	>> A(2,:) .* 2 % What is the output ? ans = 4 8 12

2 4 6 3 5 7	

Problem 17

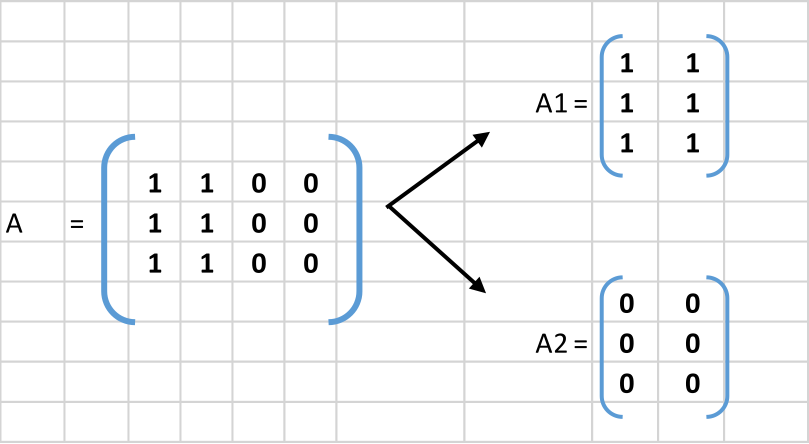
Given the matrix to the left, produce the matrix to the right: $A = \begin{bmatrix} 4 & 4 & 4 & 4 \\ 4 & 4 & 4 & 4 \\ 4 & 4 & 4 & 4 \\ 4 & 4 & 4 & 4 \end{bmatrix}$ ============> $\begin{bmatrix} 2 & 4 & 4 & 4 \\ 2 & 4 & 4 & 4 \\ 2 & 4 & 4 & 4 \\ 2 & 4 & 4 & 4 \end{bmatrix}$ >> A=ones(4).*4; % Given	Answer: >> A(: , 1)=2;

Problem 18

>> A=[1 3 5; 2 4 6; 3 5 7] A = 1 3 5 2 **4** **6** 3 **5** **7**	% Extract the following values from A : 4 6 5 7 Answer : >> A(2:3, 2:3)

Problem 19

Use programming structures to transfer the following elements of A to A1 and A2:



$$A = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix} \quad A1 = \begin{pmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{pmatrix} \quad A2 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$$

ANSWER

```
% ArrayManipulation
clc, clear
A=[1,1,0,0;
    1,1,0,0;
     1,1,0,0]
```

```
 for ii=1:1:3
     for jj=1:1:4
         if jj==1 | jj==2
            A1(ii,jj)=A(ii,jj);
         else
            A2(ii,jj-2)=A(ii,jj);
         end
     end
 end

 A1
 A2
```

Alternatively:

```
% ArrayManipulation
clc, clear
A=[1,1,0,0;
    1,1,0,0;
     1,1,0,0]

A1=A(:,1:2) % All rows, columns 1 & 2
A2=A(:,3:4  % All rows, columns 3 & 4
```

Problem 20.

Answer the following questions for the array shown here.

$$c = \begin{bmatrix} 1.1 & -3.2 & 3.4 & 0.6 \\ 0.6 & 1.1 & -0.6 & 3.1 \\ 1.3 & 0.6 & 5.5 & 0.0 \end{bmatrix}$$

(a) What is the size of c?

(b) What is the value of c(2,3)?

(c) List the subscripts of all elements containing the value 0.6.

Problem 21. Subarrays

For the array

arr1 = [1.1 -2.2 3.3 -4.4 5.5];

What is?

(a) arr1(3) Ans= 3
(b) arr1([1 4]) Ans = [1.1 -4.4]
(c) arr1(1:2:5) Ans= [1.1 3.3 5.5].

Problem 22

For the two-dimensional array

arr2 = [1 2 3; -2 -3 -4; 3 4 5]

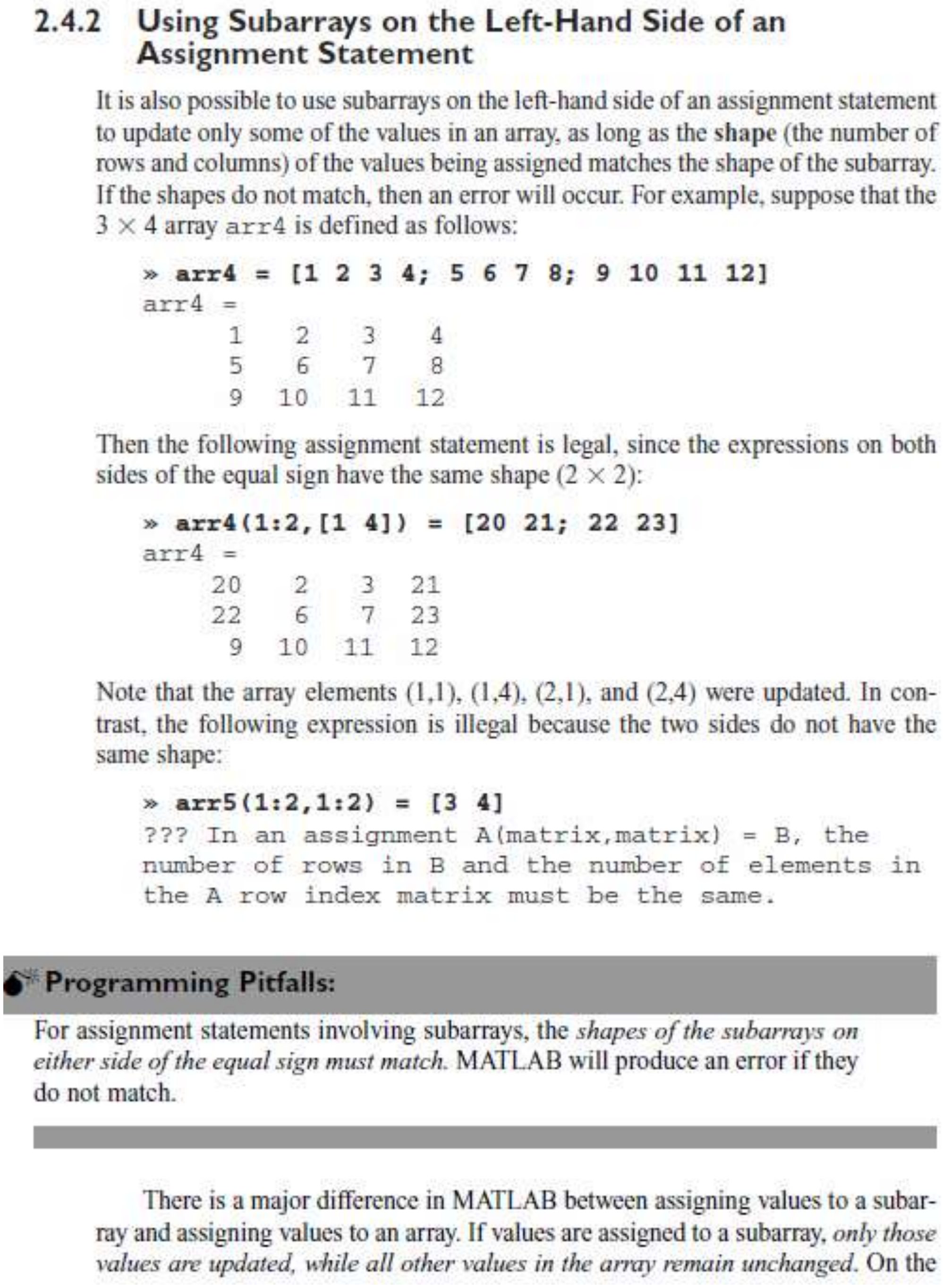
arr2=

$$\begin{bmatrix} 1 & 2 & 3 \\ -2 & -3 & -4 \\ 3 & 4 & 5 \end{bmatrix}.$$

What is?

(a) arr2(1,:) Ans=[1 2 3],

(b) arr2(:,1:2:3) Ans= $\begin{bmatrix} 1 & 3 \\ -2 & -4 \\ 3 & 5 \end{bmatrix}.$



2.4.2 Using Subarrays on the Left-Hand Side of an Assignment Statement

It is also possible to use subarrays on the left-hand side of an assignment statement to update only some of the values in an array, as long as the **shape** (the number of rows and columns) of the values being assigned matches the shape of the subarray. If the shapes do not match, then an error will occur. For example, suppose that the 3 × 4 array arr4 is defined as follows:

```
» arr4 = [1 2 3 4; 5 6 7 8; 9 10 11 12]
arr4 =
      1    2    3    4
      5    6    7    8
      9   10   11   12
```

Then the following assignment statement is legal, since the expressions on both sides of the equal sign have the same shape (2 × 2):

```
» arr4(1:2,[1 4]) = [20 21; 22 23]
arr4 =
     20    2    3   21
     22    6    7   23
      9   10   11   12
```

Note that the array elements (1,1), (1,4), (2,1), and (2,4) were updated. In contrast, the following expression is illegal because the two sides do not have the same shape:

```
» arr5(1:2,1:2) = [3 4]
??? In an assignment A(matrix,matrix) = B, the
number of rows in B and the number of elements in
the A row index matrix must be the same.
```

Programming Pitfalls:

For assignment statements involving subarrays, the *shapes of the subarrays on either side of the equal sign must match.* MATLAB will produce an error if they do not match.

There is a major difference in MATLAB between assigning values to a subarray and assigning values to an array. If values are assigned to a subarray, *only those values are updated, while all other values in the array remain unchanged.* On the other hand, if values are assigned to an array, *the entire contents of the array are deleted and replaced by the new values*. For example, suppose that the 3 × 4 array `arr4` is defined as follows:

```
» arr4 = [1 2 3 4; 5 6 7 8; 9 10 11 12]
arr4 =
      1    2    3    4
      5    6    7    8
      9   10   11   12
```

Then the following assignment statement replaces the *specified elements* of `arr4`:

```
» arr4(1:2,[1 4]) = [20 21; 22 23]
arr4 =
      20    2    3    21
      22    6    7    23
       9   10   11    12
```

In contrast, the following assignment statement replaces the *entire contents* of `arr4` with a 2 × 2 array:

```
» arr4 = [20 21; 22 23]
arr4 =
      20   21
      22   23
```

Good Programming Practice:

Be sure to distinguish between assigning values to a subarray and assigning values to an array. MATLAB behaves differently in these two cases.

2.4.3 Assigning a Scalar to a Subarray

A scalar value on the right-hand side of an assignment statement always matches the shape specified on the left-hand side. The scalar value is copied into every element specified on the left-hand side of the statement. For example, assume that the 3 × 4 array `arr4` is defined as follows:

```
arr4 = [1 2 3 4; 5 6 7 8; 9 10 11 12];
```

Then the expression that follows assigns the value one to four elements of the array.

```
» arr4(1:2,1:2) = 1
arr4 =
      1    1    3    4
      1    1    7    8
      9   10   11   12
```