

Logical arrays as mask

When a logical array is used to address another array, it extracts from that array the elements in the locations where the logical array has 1s. We can often avoid the use of loops and branching and thus create simpler and faster programs by using a logical array as a **mask** that selects elements of another array. Any of the elements not selected will remain unchanged.

The following session creates the logical array C from the numeric array A.

```
>> A = [0, -1, 4; 9, -14, 25; -34, 49, 64];  
>> C = (A >= 0);
```

The result is

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

We can use this technique to compute the square root of only those elements of A given in the previous program that are no less than 0 and add 50 to those elements that are negative. The program is:

```
>> A = [0, -1, 4; 9, -14, 25; -34, 49, 64];  
>> C = (A >= 0);  
>> A(C) = sqrt(A(C));  
>> A(~C) = A(~C) + 50;
```

The result after the third line is executed is

$$\mathbf{A} = \begin{bmatrix} 0 & -1 & 2 \\ 3 & -14 & 25 \\ -34 & 49 & 64 \end{bmatrix}$$

The result after the last line is executed is

$$\mathbf{A} = \begin{bmatrix} 0 & 49 & 2 \\ 3 & 36 & 5 \\ 16 & 7 & 8 \end{bmatrix}$$