```
function A = insertion_sortIBL(A)

n = length(A); %Sets the variable of 'n' equal to the length of the
 user-defined array

for i = 2:n %Iterating over the length of the array. We can't start at
 1 because there is no prior integer to compare it to.
    j = i; %Sets a new variable 'j' equal to the for index 'i.' This
 is necessary so that the while loop doesn't interfere with the for
 loop.
    while (j > 1) && (A(j-1) > A(j))
        placeholder = A(j);
        A(j) = A(j-1);
        A(j-1) = placeholder;
        j = j-1;
        %While the for loop index (the position of the integer) is
 greater
        %than 1 and the current integer value is smaller than the one
 that
        %was before it in the list, the loop repeats. To swap the
 positions
        %of the integers, the placeholder expression is used. We did
 this
        %so that the original value was not lost in the switching
 process.
        %The insertion algorithm progressively shifts the larger
 integer
        %values to the left until the array is sorted.
    end %Ends the while loop.
end %Ends the foor loop

disp(A) %Displays the sorted array

end %Ends the function

Not enough input arguments.

Error in insertion_sortIBL (line 3)
n = length(A); %Sets the variable of 'n' equal to the length of the
 user-defined array
```

Published with MATLAB® R2021a