```
% Function to return the specified number of largest or smallest blobs
 in a binary image.
% If numberToExtract > 0 it returns the numberToExtract largest blobs.
% If numberToExtract < 0 it returns the numberToExtract smallest
 blobs.
% Example: return a binary image with only the largest blob:
%   binaryImage = ExtractNLargestBlobs(binaryImage, 1)
% Example: return a binary image with the 3 smallest blobs:
%   binaryImage = ExtractNLargestBlobs(binaryImage, -3)
function binaryImage = ExtractNLargestBlobs(binaryImage,
 numberToExtract)
try
 % Get all the blob properties.  Can only pass in originalImage in
 version R2008a and later.
 [labeledImage, numberOfBlobs] = bwlabel(binaryImage);
 blobMeasurements = regionprops(labeledImage, 'area');
 % Get all the areas
 allAreas = [blobMeasurements.Area];
 if numberToExtract > 0
  % For positive numbers, sort in order of largest to smallest.
  % Sort them.
  [sortedAreas, sortIndexes] = sort(allAreas, 'descend');
 elseif numberToExtract < 0
  % For negative numbers, sort in order of smallest to largest.
  % Sort them.
  [sortedAreas, sortIndexes] = sort(allAreas, 'ascend');
  % Need to negate numberToExtract so we can use it in sortIndexes
 later.
  numberToExtract = -numberToExtract;
 else
  % numberToExtract = 0.  Shouldn't happen.  Return no blobs.
  binaryImage = false(size(binaryImage));
  return;
 end
 % Extract the "numberToExtract" largest blob(a)s using ismember().
 biggestBlob = ismember(labeledImage, sortIndexes(1:numberToExtract));
 % Convert from integer labeled image into binary (logical) image.
 binaryImage = biggestBlob > 0;
catch ME
 errorMessage = sprintf('Error in function ExtractNLargestBlobs().\n
\nError Message:\n%s', ME.message);
 fprintf(1, '%s\n', errorMessage);
 uiwait(warndlg(errorMessage));
end
```

Published with MATLAB® R2016b