Cody Problem 30. Sort a list of complex numbers based on far they are from the origin.

Given a list of complex numbers z, return a list zSorted such that the numbers that are farthest from the origin (0+0i) appear first.

So if z is

```
% z = [-4 6 3+4i 1+i 0]
```

then the output zSorted would be

```
% zSorted = [6 3+4i -4 1+i 0]
```

Scratch Pad

```
z = [-4 6 3+4i 1+i 0]
```

```
z = 1×5 complex
  -4.0000 + 0.0000i   6.0000 + 0.0000i   3.0000 + 4.0000i   1.0000 + 1.0000i ···
```

```
zSorted = complexSort(z)
```

```
zSorted = 1×5 complex
   6.0000 + 0.0000i   3.0000 + 4.0000i  -4.0000 + 0.0000i   1.0000 + 1.0000i ···
```

Solution

```
function zSorted = complexSort(complexArray)
    Ds = zeros(length(complexArray), 1);  % Preallocate Ds
    % Calculate distances and store them in Ds
    for k = 1:length(complexArray)
        z = complexArray(k);          % Get the k-th complex number
        real_part = real(z);          % Access the real part of z
        imaginary_part = imag(z);     % Access the imaginary part of z
        Ds(k) = sqrt(real_part^2 + imaginary_part^2);  % Calculate the
distance
    end

    % Sort complexArray based on the calculated distances
    [~, idx] = sort(Ds, 'descend');
    zSorted = complexArray(idx);
end
```