MATLAB Programming NCTU Fall 2022 LAB#02 2022/9/27

Exercises in logical indexing and program control.

1. Draw a filled circle (an exercise for logical indexing; use no loop)
   (a) Make a square matrix `A` of size `n`x`n`. Make n an odd number.
   (b) Compute the "distances" of all the elements to the center element. Store these in a "distance matrix" `D`, also of size `n`x`n`. Note: Function `meshgrid` is useful here.
   (c) For a given radius `r` (`r` > 0; `r` can be a floating-point number), set `A(ii,jj)` to 1 if `D(ii,jj)<r`, and 0 otherwise. Then just print out `A`. Example below for `n`=7 and `r`=2.5:

```
     0     0     0     0     0     0     0
     0     0     1     1     1     0     0
     0     1     1     1     1     1     0
     0     1     1     1     1     1     0
     0     1     1     1     1     1     0
     0     0     1     1     1     0     0
     0     0     0     0     0     0     0
```

2. Pascal Triangle
   Note: You can use <u>**one level**</u> of loop.
   For a given integer `n`>0, print out the Pascal triangle with `n` levels. Example for `n`=5:

```
     1
     1     1
     1     2     1
     1     3     3     1
     1     4     6     4     1
```

   Store the values of each level in a vector, which can be computed and printed (use `disp`) in one statement.
   Hint: If the vector in one level is `v`, then the vector in the next level is the sum of `v` and its shifted version. Create the shifted version by padding a zero at the front or the back of `v`, such as `[0 v]` and `[v 0]`.

3. Find the unique elements of a numerical vector without using the function `unique`.
   Note: Sort the vector first. The returned vector, containing only unique values, should also be sorted.
   Example: If the initial vector is `[3 5 2 2 3 7 1]`, the output should be `[1 2 3 5 7]`.
   <u>Implementation #1</u>: Use <u>**one level**</u> of loop to select elements that are different from their immediate predecessors. Append each selected element to the output vector. The output vector should be initialized with the first element after sorting (why?).
   <u>Implementation #2</u> (no loop used): Use logical indexing to select such elements, by comparing the vector (after sorting) with its shifted version.
   You only need to do one of the two implementations.