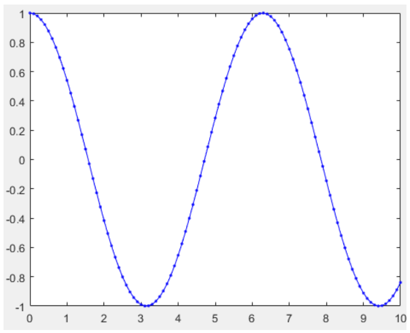
MATLAB Programming NCTU Spring 2017 Exam#1 2017/4/21

You need to write a m-file for each problem. Name your m-files **P1.m**, **P2.m**, etc.

You can consult the lecture slides, your own class notes, MATLAB documentation, and other printed material you can bring with you, but no other material is allowed. You are not allowed to talk with one another or use your own 3C devices or USB drives. You can use any function that we have covered in the class unless noted otherwise, or any standard MATLAB functions you can find. However, you cannot use toolbox functions that are not covered in the class.



1. [25%] Write a function **P1** that computes the sum: $f_n(x) = \sum_{k=0}^{n} \frac{(-1)^k x^{2k}}{(2k)!}$.

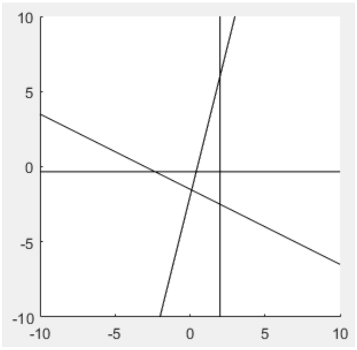
The function takes two inputs, x and n. Here n is a non-negative scalar integer and x is a vector. Note: The maximum points you get are reduced by 5 for each layer of loop used.

Just so that you can check your results, the following statements should generate the plot here:

```
x=0:.1:10;  y=P1(100,x);  plot(x,y,'-b.');
```

2. [15%] Write a function **P2** that takes an input n, a positive odd integer, and output a nxn diagonal matrix where each element gives its Manhattan distance from the center. Note: Use no loops. Example output matrix for n=5:

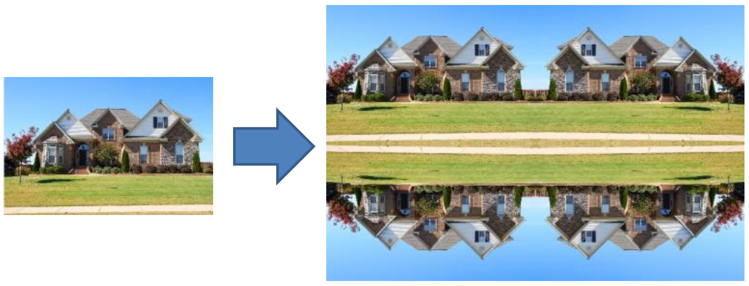
```
4 3 2 3 4
3 2 1 2 3
2 1 0 1 2
3 2 1 2 3
4 3 2 3 4
```



3. [25%] Write a function **P3** that takes one input matrix W. Each row of W is a 3-element vector [a b c] that represents a 2-D line given by ax+by+c=0. Draw all the lines in a single 2-D plot. (Set the range to -10~10 for both x and y axes.) You can use a layer of loop.

Example plot for `P3([1 2 3; 2 0 -4; 0 3 1; -4 1 2])`:

4. [15%] Write a function **P4** that takes an image im as its only input. In your function, create images by flipping im in the horizontal, vertical, and both directions. Display their concatenated image. Note: im may be a gray-scale or RGB image. Note: Use no loops. An example is given here:



5. [20%] Write a function **P5** that takes a single string as input. This string contains a series of words separated by white spaces. The function should generate two outputs. The first output is the maximum length of those words, and the second output is a cell array of strings containing those words with the maximum length. You are allowed a layer of loop here. Below is an example:

```
 [m, s] = P5('one two three four five six seven eight')
m =
      5
s =
    'three'    'seven'    'eight'
```