Introduction to Scientific Computing, PSCB57, Fall 2016
Assignment 3
Linear Least Square Fit

- The deadline for this assignment is Friday, October 7th, 5pm.
- This assignment comes in multiple parts. Submit all your answers in one file.
- Only submit the functions you wrote, i.e. no other code such as print statements that you might have used to test your submission.
- Do not use any packages or libraries except the ones specifically mentioned below.
- The entire submission has to be 70 lines of code or less (including comments).
- You must submit the assignment electronically at `http://rein.utsc.utoronto.ca/submit/`. On the same website, you can run an automated test of your program before submitting it. The username is `pscb57`, the password is `2016`.
- Your code needs to pass all tests in 4 seconds or less.
- You must be present at the tutorial in the following week to take a quiz. If you do not show up or fail to pass the quiz, your assignment will be marked as 0% even if it was correct.

Part 1

Write a function `s(A,v)` which takes as arguments a square matrix A with arbitrary dimensions $N \times N$ and a vector v with the same number of dimensions, N (in python A is represented by a list of lists). Your function should return the vector x which is the solution to the linear system of equations $A \cdot x = v$. You may assume the matrix A is invertible.

You may solve this part in one of the two following ways.

- You can implement the Gaussian elimination method that we derived in the lectures. However, if you do that, you need to also implement some pivoting algorithm that ensures your function is well behaved and does not run into floating point issues as your function needs to work for arbitrary invertible matrices A.
- You can use the `linalg.solve()` function from the numpy package. If you choose to use the numpy package, you may import the solve function as follows, but not any other function:

```
from numpy.linalg import solve
```

The first option has the advantage that you do not need to install any packages on your computer (which can be a hassle). The second option has the advantages that once you successfully imported the numpy package, this part of the assignment is very easy!

Part 2

Write a function `lsf(x,y,f)` to perform a linear least square fit of the function f to the data given by x and y. The first two arguments of `lsf` are the vectors x and y, representing the data. Each of those vectors has dimensions N. The third argument of `lsf` corresponds to the function f that you want to fit. Your program should be generic enough so that it works for arbitrary linear functions. The function `f(x)` takes one argument, the scalar x, and returns a vector with dimensions M (see example below).
Your function should also return a vector with dimensions M. You have to determine M from the return value of f. The return value of your function should be the coefficients of the linear least square fit, a_i.
As an example and a test which you can use while writing your program, we will look at the mean global temperature anomaly. This dataset is often used in relation to global warming. The first column represents the calendar year, the second column represents the average temperature on Earth compared to 1900. You can also download the dataset from the course website.

```
1980 0.135     1989 0.245     1998 0.835     2007 0.909
1981 0.317     1990 0.492     1999 0.561     2008 0.695
1982 -0.002    1991 0.397     2000 0.484     2009 0.731
1983 0.331     1992 0.093     2001 0.681     2010 0.901
1984 -0.045    1993 0.175     2002 0.778     2011 0.695
1985 -0.033    1994 0.343     2003 0.770     2012 0.751
1986 0.120     1995 0.592     2004 0.674     2013 0.791
1987 0.252     1996 0.227     2005 0.882     2014 0.762
1988 0.375     1997 0.518     2006 0.810
```

Try fitting the data to a linear and quadratic model of the global temperature:

$$f_1(x) = a_0 + a_1 \cdot x \quad (1)$$

$$f_2(x) = a_0 + a_1 \cdot x + a_2 \cdot x^2 \quad (2)$$

For f_1, your algorithm should return a two dimensional vector with the best fit values for a_0 and a_1. For f_2, your function should return a three dimensional vector containing (a_0, a_1, a_2). You would implement f_1 and f_2 in the following way in python:

```
def f1(x):
    return [1., x]
def f2(x):
    return [1., x, x*x]
```

Thus, these functions return the individual terms in the functions f_1 and f_2, but without the linear coefficients a_i (we are trying to find those!). In this specific example, we have $N = 35$, and $M = 2$ for f_1 and $M = 3$ for f_2 and the solutions are:

$$\text{for } f_1: \quad a_0 = -48.69734,\ a_1 = 0.024631 \quad (3)$$

$$\text{for } f_2: \quad a_0 = -1015.407,\ a_1 = 0.992819,\ a_2 = -0.000242410 \quad (4)$$

Can you use your fits to predict the global temperature in the year 2050?

Note: Only submit the functions `s(A,x)` and `lsf(x,y,f)`. Do not submit the above example or any of the test data. Keep in mind that you need to write your function general enough so that it works with arbitrary datasets x, y and arbitrary linear functions f.