Instructions:
Name your script `hw7.py` and submit it on CCLE. Add comments to each function.

Problem 1:
The Pearson correlation coefficient, r, is a commonly used measure of linear correlation between two variables $\vec{x}$ and $\vec{y}$, taking the forms of two sample datasets $\vec{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$ and $\vec{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$. The usual definition does not use linear algebra and looks like:

$$r = \frac{\sum_{i=1}^{n}(x_i - \mu_x)(y_i - \mu_y)}{\sqrt{\sum_{i=1}^{n}(x_i - \mu_x)^2}\sqrt{\sum_{i=1}^{n}(y_i - \mu_y)^2}}.$$

If we "centralize" the vectors $\vec{x}$ and $\vec{y}$ around the means, or in other words, consider the deviation vectors $\vec{x_c} = \begin{pmatrix} x_1 - \mu_x \\ x_2 - \mu_x \\ \vdots \\ x_n - \mu_x \end{pmatrix}$ and $\vec{y_c} = \begin{pmatrix} y_1 - \mu_y \\ y_2 - \mu_y \\ \vdots \\ y_n - \mu_y \end{pmatrix}$ then we have

$$r = \cos(\theta) = \frac{\vec{x_c} \cdot \vec{y_c}}{\|\vec{x_c}\|_2 \times \|\vec{y_c}\|_2}.$$

This corresponds to thinking of the centralized datasets as two vectors in n-dimensional space, where each dimension corresponds to an observation. These two vectors are correlated if they point in the same direction, i.e. if a positive deviation from the mean of one vector in some direction results in a positive deviation of the other vector. (And same for negative deviations.) The two variables are anti-correlated if they point in opposite directions. Remember that the cosine of an angle is 1 when the angle is 0, and -1 when the angle is 180°.

Write a function named `corrcoeff(x,y)` that takes two vectors of length `n` as input (representing two variables at `n` observations), and outputs the correlation coefficient between them.

Test case:

```
x = np.array([3,4,6,1,2,3])
y = np.array([2,3,1,4,1,2])
corrcoeff(x,y) should return -0.51318
```

Problem 2:
Use integration in SymPy to write a function named `normalcurve(a,b)` that takes as input two boundaries a, b, and returns the probability that the a standard normal random variable falls in the interval between a and b.

Problem 3:
Write a function named `balance(eq)` that balances chemical equations. So, it takes as input strings of the form `"H2+O2=H2O"` into `"2H2+O2=2H2O"`. This function does not need to account for parenthesis like Pb(OH)4 or Pb(SO4)2.