Homework 13

Problem 1

Part a

- The homework includes a file `units.py`. The file should be in the same folder as your HW13.ipynb file that you are working on (this file).
- Show the contents of that file in your notebook by typing `%load units.py` in a cell and running it.

In [ ]:

Part b

- Extend the `units.py` module in a new `ext_units.py` file with the leading line `from units import *`
- Add functions to convert from F to R and R to F.
- Correct the conversions from C to K and from K to C by overriding those functions.
- Test the conversion functions by computing the freezing point of water at $T_f = 0^o\ C$ in F, K, and R.

In [ ]:

Part c

- Import the `ext_units.py` module and use it in the function you write below.
- Define a function that returns the ideal gas pressure given temperature with n=1 mole and V=1 m^3.
  - The function should look like this: `def P_ig(T, Tu) :`,
  - where, T is the Temperature and Tu is the units of Temperature.
- Hint: Convert all input units to K, then compute pressure (in Pa).
- Test the function by computing the pressures at temperatures of $0^o\ C$, $32^o\ F$, and $491.67\ R$.

In [ ]:

Problem 2

Part a

- Write a class for computing thermodynamic properties in a cell below.
- Call the class "thermo"
- Include an `__init__(self, species)` function that sets the gas constant Rgas = 8314.46 J/kmol*K.
  - Use kmol instead of gmol because kg is the SI unit of mass, not gm.
  - The **init** function should open the file thermoData.yaml included with the homework. Use this code to open the file:

```
import yaml
with open("thermoData.yaml") as yfile :
   yfile = yaml.load(yfile)
```

- Also in `__init__` Make two arrays that are members of the class called a_lo, and a_hi.
- Get these arrays from the yaml file using something like `a_lo = yfile[species]["a_lo"]`, where "species" is the string passed as an argument to **init**. When you create an instance of the class, you should give a string argument that is one of the species in the HW13P2.yaml file.
- The two arrays work in two separate temperature ranges: a_lo is for $300

Part b

- Write thermo class functions: get_cp that computes the heat capacity in J/kmol*K. The functions should take temperature in Kelvin as an argument.
- Make sure you use the right set of coefficients for the right temperature range.
- The equation is given by:

$$\frac{c_p}{R_g} = a_0 + a_1 T + a_2 T^2 + a_3 T^3 + a_4 T^4$$

Part c

- Instantiate the class for species CH4. Call the object `t_CH4`
  - Like this: `t_CH4 = thermo("CH4")`
- Make a plot for cp versus temperature. Let temperature vary from 300 K to 3000 K. Use at least 100 points.
- Hints:
  - Make an array of temperatures and heat capacities (cp)
  - Write a loop over the number of points: `for iT in range(npts) :`
  - In the loop call the functions with something like: `cp[iT] = t_CH4.get_cp(T[iT])`

In [ ]: