Part A:
Hierarchy:

		Temp range	
		Temp input	
		Find temp range	
volume	Internal energy	enthalpy	entropy
Volume interpolation	Internal energy interpolation	Enthalpy interpolation	Entropy interpolation

```
# By submitting this assignment, I agree to the following:
#   "Aggies do not lie, cheat, or steal, or tolerate those who do."
#   "I have not given or received any unauthorized aid on this assignment."
#
# Names:        Mariana Sanchez
#               Aaron Garcia
#               Dhruv Manihar
#               Ethan Nguyen
# Section:      514
# Assignment:   Lab 8
# Date:         29 10 2021
#
# temp then v,u,h,s in that order
temp = [0,20,40,60,80,100,120,140,160,180,200,220,240,260]

volume = [0.0009977, 0.0009996, 0.0010057, 0.0010149, 0.0010267,
        0.0010410, 0.0010576, 0.0010769, 0.0010988, 0.0011240,
        0.0011531, 0.0011868, 0.0012268, 0.0012755]

internal_e = [0.04, 83.61, 166.92,  250.29, 333.82, 417.65,
            501.91, 586.80, 672.55, 759.47, 847.92, 938.39,
            1031.6, 1128.5]

enthalpy = [5.03, 88.61, 171.95, 255.36, 338.96, 422.85,
          507.19, 592.18, 678.04, 765.09, 853.68, 944.32,
          1037.7, 1134.9]

entropy = [0.0001, 0.2954, 0.5705, 0.8287, 1.0723, 1.3034, 1.5236,
         1.7344, 1.9374, 2.1338, 2.3251, 2.5127, 2.6983, 2.8841]
```

```
#inputs
user_temp = float(input("Enter a temperature between 0 and 260 deg C: "))
print("Properties at {} deg C are:".format(user_temp))

#conditions
for i in range(len(temp)):
    one = float(temp[i])
    two = float(temp[i+1])
    if user_temp >= one and user_temp <= two:
        break

#volume (v) (aaron)
vi= volume[i]
vf= volume[i+1]
ti= one
tf= two
given_t= user_temp
v = ((vf-vi)/(tf-ti))*(given_t-tf)+vf

#internal energy (u) (dhruv)
ei= internal_e[i]
ef= internal_e[i+1]
ti= one
tf= two
given_t= user_temp
u = ((ef-ei)/(tf-ti))*(given_t-tf)+ef

#enthalpy (h) (mariana)
thi= enthalpy[i]
thf= enthalpy[i+1]
ti= one
tf= two
given_t= user_temp
h = ((thf-thi)/(tf-ti))*(given_t-tf)+thf

#entropy (s) (ethan)
tri= entropy[i]
trf= entropy[i+1]
ti= one
tf= two
given_t= user_temp
s = ((trf-tri)/(tf-ti))*(given_t-tf)+trf
```

```
#print statements
print('Specific volume (m^3/kg): {:.7f}'.format(v))
print('Specific internal energy (kJ/kg): {:.2f}'.format(u))
print('Specific enthalpy (kJ/kg): {:.2f}'.format(h))
print('Specific entropy (kJ/kgK): {:.4f}'.format(s))
```

Part B - variable list

temp = list of temperatures in data
volume = list of volumes in data
internal_e = list of internal energy in data
enthalpy = list of enthalpy numbers in data
entropy = list of entropy numbers in data
user_temp = users input for the temperature

    Used in for loop
    one = number taken from temp list
    two = number taken from temp list adding one for next value

vi = initial volume taken from list
vf = final volume taken from list starting at next value
ti = one (first temp from list)
tf = two (second temp from list)
given_t = user_temp (new variable to use the user input)
v = the interpolate for these numbers in volume
ei = initial number from internal energy list
ef = final number for internal energy list adding one
ti = one (first temp from list)
tf = two (second temp from list)
given_t = user_temp (new variable to use the user input)
u = the interpolate for these numbers in volume
thi = initial number from enthalpy list
thf = final number for enthalpy list adding one
ti = one (first temp from list)
tf = two (second temp from list)
given_t = user_temp (new variable to use the user input)
h = the interpolate for these numbers in volume
tri = initial number from entropy list
trf = final number for entropy list adding one
ti = one (first temp from list)
tf = two (second temp from list)
given_t = user_temp (new variable to use the user input)
s = the interpolate for these numbers in volume

Part C/D Test cases:

Temp	Volume	Internal energy	enthalpy	entropy
0	.0009977	.04	5.03	.0001
260	.0012755	1128.50	1134.90	2.8841
50	.0010103	208.60	213.66	.6996
10	. 0009987	41.83	46.82	.1477
20	.0009996	83.61	88.61	.2954
30	.0010026	125.26	130.28	.4330
40	.0010057	166.92	171.95	.5705
70	.0010208	292.06	297.16	.9505
67	.0010190	279.53	284.62	.9140
54	.0010121	225.28	230.34	.7512

Outline:

Assign Lists:
List of temps
List of volumes
List of internal_energy
List of enthalpy
List of entropy

Input:
Temp input

Loop:
Loop to determine the two temperature values user input is between

Volume Calculations:
Gathers two volumes values from temperature points and interpolates for user specific value
Internal Energy Calculations:
Gathers two internal energy values from temperature points and interpolates for user specific value

Enthalpy Calculations:
Gathers two enthalpy from temperature points and interpolates for user specific value

Entropy Calculations:
Gathers two entropy from temperature points and interpolates for user specific value

Print Statements:
Prints all values to decimal precision-

- Volume - 7
- I.E - 2
- Enthalpy - 2
- Entropy - 4

Part E:
Separated jobs:

#master list (Aaron)

```
#volume (v) (aaron)
vi= volume[i]
vf= volume[i+1]
ti= one
tf= two
given_t= user_temp
v = ((vf-vi)/(tf-ti))*(given_t-tf)+vf

#internal energy (u) (dhruv)
ei= internal_e[i]
ef= internal_e[i+1]
ti= one
tf= two
given_t= user_temp
u = ((ef-ei)/(tf-ti))*(given_t-tf)+ef

#enthalpy (h) (mariana)
```

```
thi= enthalpy[i]
thf= enthalpy[i+1]
ti= one
tf= two
given_t= user_temp
h = ((thf-thi)/(tf-ti))*(given_t-tf)+thf

#entropy (s) (ethan)
tri= entropy[i]
trf= entropy[i+1]
ti= one
tf= two
given_t= user_temp
s = ((trf-tri)/(tf-ti))*(given_t-tf)+trf
```

Part G:

Summary:

Coding by following the top down system and splitting up the code for every person made each individual part easier to make, however it did present some issues when we got to the coding part, we had to create a for loop to find the position in the list that each input would fall under. This system of coding could be a bad idea if one part doesn't work or if each part is done differently and therefore needs different conditions met.