**Winter 2016 - Cmpt 103**

**Section 41 - Lab 10 mark sheet**

**Student name:** Metehan Dagsuyu  **Total: 90 marks**

**Programming style: [8 marks]**

* Reasonable code spacing although extract\_unique\_elements might benefit from a bit more spacing
* Good comments throughout

**Question 1: sum\_to [26 marks]**

* Function header for sum\_to: **-1**
  + The syntax should start with ‘total\_sum =’ as that is what is returned.
  + The return value should start with ‘total\_sum: sum of the first n reciprocals’.
* Function worked properly when tested

**Question 2: cantor\_set [29.5 marks]**

* Function header for main function: **-1**
  + Missing parameters and return value. You have to explicitly write None even if they aren`t present so I know that you didn`t forget them by accident.
* Function header for add\_line: **-1**
  + The syntax should start with ‘line =’ as that is what is returned.
  + The return value should start with ‘line:’ to become ‘line: a line object’.
* Function header for cantor\_set: **-0.5**
  + The syntax includes win but the actual function call doesn`t.
* Very colorful lines!
* Function cantor\_set was named incorrectly as drawCantor which crashed my test program due to the different function name: **-1**
* Function worked properly when tested

**Question 3: extract\_unique\_elements [26.5 marks]**

* Function header for extract\_unique\_elements: **-0.5**
  + The syntax should start with ‘unique\_values =’ as that is returned
* The code works correctly when you press the green arrow after each run as it clears your global variable back to its initial state. Otherwise, it continues to grow the global variable which is not what you want.
* extract\_unique\_elements does not properly perform the recursion:
  + Does not create an empty set to hold the final result locally. It was done globally which should not be the case. Move the line all\_elements = [] inside of the loop and ensure it is not global anymore: **-1**
  + Does not properly accumulate all of the values from each recursive call to produce a final set. As you are using lists, you need all\_elements += in front of the extract\_unique\_elements(each) function call: **-4**
* This is a revised version of your code:

def extract\_unique\_elements(nested\_lists):

all\_elements = []

if isinstance(nested\_lists, list) or isinstance(nested\_lists, tuple):

for each in nested\_lists:

all\_elements += extract\_unique\_elements(each)

else:

all\_elements.append(nested\_lists)

unique\_elements = set(all\_elements)

return unique\_elements