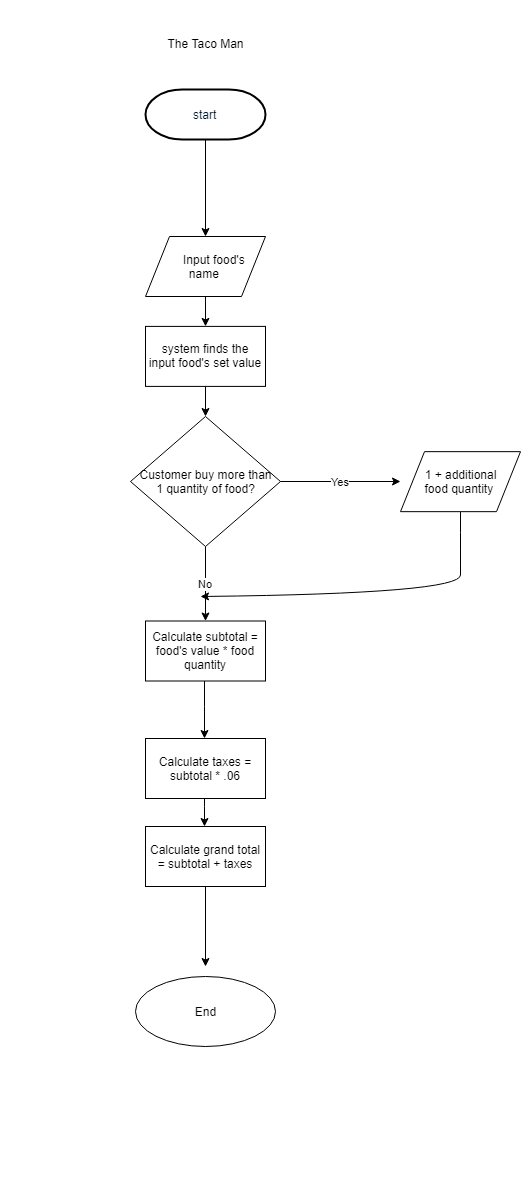
#1 Flowchart:



#1 Pseudocode:

* Start
* OBTAIN the measurements for the wall’s height and its width in feet
* Write wall\_Height value from the measurement OBTAINED for the wall’s height.
* Write wall\_Width value from the measurement OBTAINED for the wall’s width.
* CALCULATE the square feet for the area of the wall by multiplying the wall’s height with its width
* Write wall\_Area value from the calculation
* ASSUME one gallon of paint covers 350 square feet
* CALCULATE the amount of paint needed by dividing the wall’s square feet by 350 square feet
* Write paint\_Needed value from the calculation
* Round up the amount of paint needed to find the amount of can(s) needed
* Write can\_Needed value after rounding up
* After getting the value of can(s) needed, go to Home Depot’s paint section
* Once at the paint section, call wife and ask what paint color she wants the wall to be (Input)
* Find Home Depot’s set price for the requested color (Dictionary)
* Grab the amount of 1-gallon paint can(s) for the color requested to get
* Pay the amount of 1-gallon paint can(s) multiplied by the set price for the requested color
* Stop

#2:

* Which one did you find easier?

I found the Pseudocode to be easier than the flowchart because I was able to create the process in my own words. It was also helpful to visualize the process as a list as I did in my Pseudocode.

* What challenges did you face creating pseudocode or a flowchart (or both)?

I found the flowchart to be challenging because of the verbiage and the shapes needed to describe my process. I found the Pseudocode easier because it reminded me of using Task Decomposition like in the ZYbooks 1.2.

* Given that functioning programs are what is actually run by users, how would skipping the creation of pseudocode and flowcharts impact the program and/or the programming process?

I believe skipping the creation of pseudocode and flowchart could impact the program process because they are like a manual created by the programmer before implementing the coding. These are used to show the steps and direction the programmer wants the process of the coding to go by either creating a basic list like the pseudocode or a visual direction chart like the flowchart.