|  |  |
| --- | --- |
| **Goal for exercise – “Introduction to Python and Raspberry Pi”**  Goal for this exercise was to familiarize myself with coding/working on a raspberry pi, as well as learning how to code more advanced concepts in python such as Finite State Machines. | |
| **What do you know about Pi computers and Python?**  I have worked with python and open cv on the raspberry pi before. I have spent a good amount of time with a Pi 2. Specifically I have worked with the GPIO pins, I2c,libraries and pip. I am fairly familiar with the command line commands on the Pi, just don’t have them memorized. I have don’t a lot of coding in Python, its ease of use made it my preferred language for many years. | **List all resources and what specifically you used or learnt from that resource to complete the challenge exercises.**  I was previously in familiar with NumPy. By using their technical documents I was able to greatly reduce the difficulty of exercise 1. Specifically I looked up their technical documentation on their Max, Min, Size, Where, Flip, Sort and Argsort functions.  It has also been a while since I’ve coded in Python, so their were a few small things I forgot including the exact syntax for “for” loops and how to handle exceptions with try. I found answers to my questions through google, specifically on forum sites. I also wanted to clean up my printing so I did some researching on printing special characters and tabs in python.  The included FSM examples gave me reference on how I could code my own FSM in Python. |
| **Compile a list of all documentation created. Provide file name and a short description of that file**.  All of my documentation is within “assignment1\_documentation.pdf”. This file contains both answers to the assignment questions and the code for both exercise 1 and 2. The header of the exercise code provides both run instructions and the basics of how it works. The foot contains the various online resources I used to write the code. | |
| **Provide an example of something that you would do differently or you could improve upon during the course of this exercise.**  I feel like I could do a better job handling the exceptions in excise 1. While I do printout my own error message, python is still the one automatically halting the application. I would like to implement a system where the user can type in the file name, and have the choice of typing in another if it is invalid. Overall adding user input to exercise 1 would beneficial. Write now you would have to change the code if you want to switch from even to odd sorted for example. | |
| **On a scale of 1-5, what is your comfort level with Pi computers and Python after going through this exercise? (1 being least comfortable and 5 being most comfortable).**  1 2 3 4 5 | |