Project Kotsen READme Team Caz Kotsen

1. **Team name:**
   Team Kotsen
2. **Team Members and NetID:**
   Caz Kotsen, ckotsen
3. **Project Type:**
   Program 2, K tape TM
4. **Success:**
   I would say the program is very successful
5. **Approximate time:**
   It took me about 4 hours in total to complete. Generating input scripts was time consuming.
6. **Github Link:**
   https://github.com/cjkot1122/Project2
7. **List of Files:**
   Code file names: TuringMachine.py —--
   Test files were inputs1.txt, inputs2.txt, inputs3.txt with output files output1.txt, output2.txt, output3.txt
8. **Programming languages and libraries used:**
   I used python with subprocess, os, and sys
9. **Data Structures:**
   Lists, file objects, ints, strings, and dictionaries were all used.
10. **Operation of Code:**
    The Turing Machine code simulated a TM solving the given input files. The code followed general L, R, and stay rules. It maintained a max total step size to stop the code from running forever. The program very simply parses the inputs, finds matches, and runs the simulation. I added some messages to show why the code would halt, and ultimately had the steps printed in an output file.
11. **Test Cases:**
    Overall, I just used test cases that were very all encompassing, I wanted to make sure all edge cases were covered and different issues could be handled such as varying input size. These test cases helped me to find issues in my code while I worked and by the end showed me my code worked correctly at least for the test cases I could think of.
12. **Code Development:**
    For this particular project I tried to develop the code all at once in order to not lose myself in the process. Generating the input was the first step and with that fresh on my mind I was able to make a baseline version of the sim code. I worked through debugging errors and then moved on to thinking of inputs with particular edge cases that I need to be accounted for. After that it was just structuring output files and doing final checks.
13. **Results:**
    I was fairly happy with the results of my project. I think I added in a solid messaging and output system in order to ensure my TM worked as intended. The coding part was not the most difficult, it was more so just making sure I remembered all the different types of problems the TM could potentially see. It was a lot of tracking back to add in little details

that helped with edge cases. Overall, I feel my code reached a point where it covers any type of problem we saw in class and will swiftly reject invalid inputs.

14. **Team Organization:**
    Alone
15. **What to do differently next time:**
    If I were to do it again differently, I might have tried to get a jump on it earlier in the semester. I ended up having interviews scheduled for this week and had to prioritize my time, hence the extension.