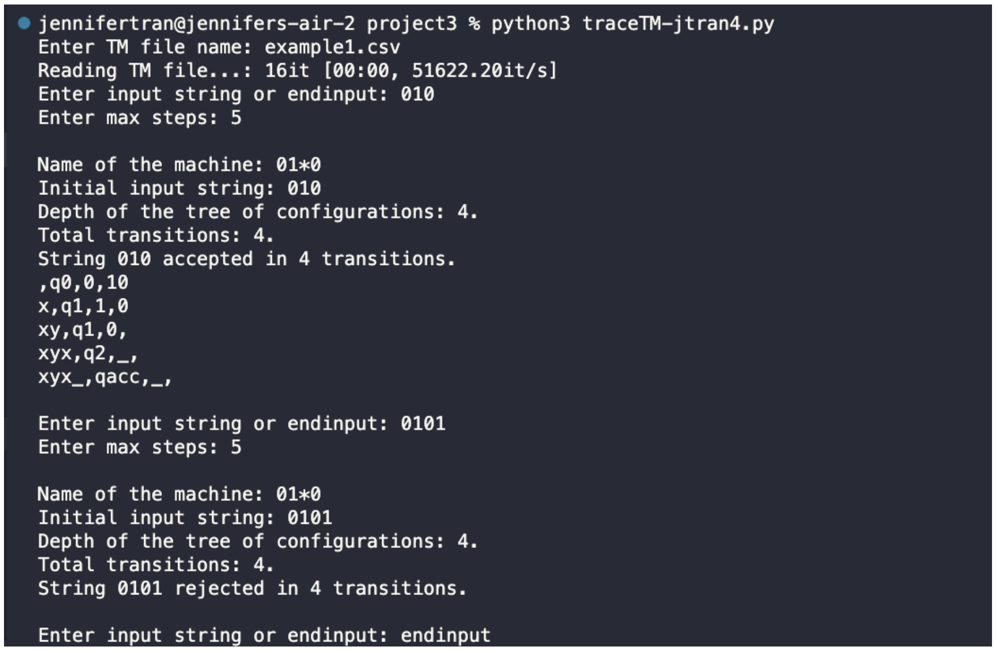
1. The team name.
    - jtran4

2. The members of the team.
    - Name: Jennifer Tran
    - netID: jtran4

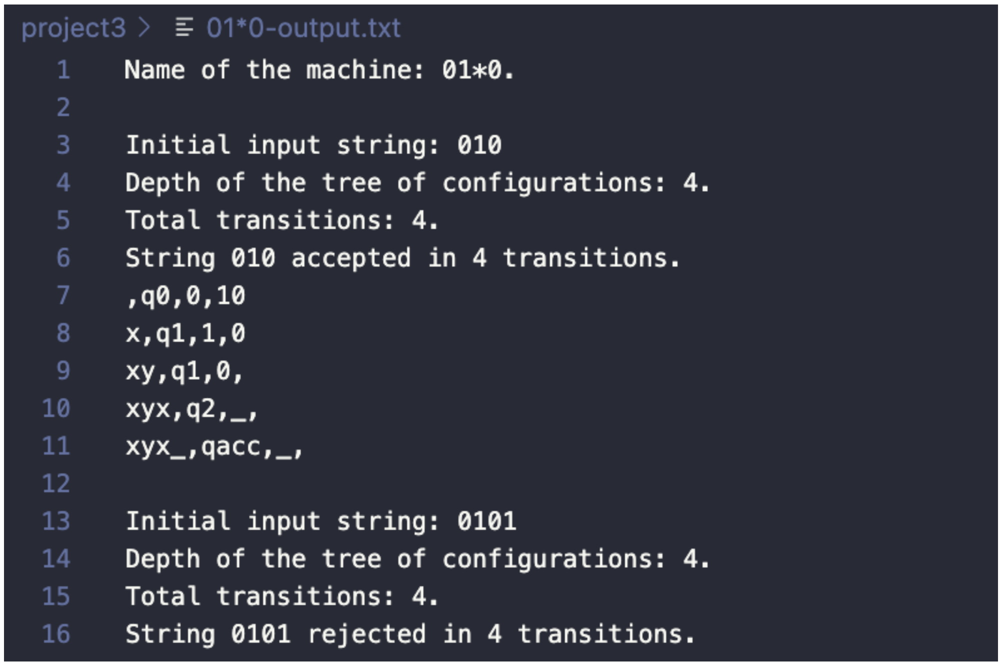
3. Approximately how much time was spent in total on the project, and how much by each student.
    - Approximately 8.5 hours were spent on this project in total.
        - 4 hours were spent creating the code from scratch.
        - 2 hours were spent in office hours getting help with the code.
        - 1.5 hours were spent on creating the test cases for the code
        - 1 hour was spent on writing the reflection for the “readme-jtran4” and “teamwork-jtran4” files.

4. A description of how you managed the code development and testing. Use of github in particular is a particularly strong suggestion to simplifying this process, as is some sort of organized code review.
    - Planning:
        - Review lectures about the non-deterministic Turing Machine
        - Understand the input and output format for the program
        - Create the Turing Machine (TM) and Tape class objects
        - Create the function to read the Turing Machine from the CSV input and store all needed data in the TM object
        - Implement trace function
        - Test the program

    - Overview of the program:
        - The traceTM-jtran4.py program will ask the user to enter the file name of the Turing Machine. After reading the Turing Machine, it will continuously ask the user to input the string and the maximum steps (or maximum transition). Every time the program reads in an input string and a max step, it will print the output to the screen.
        - The program will terminate when the input string is “endinput”. The program will then record all the computed strings and their outputs in the text file with the name [name of machine]-output.txt.
        - For example, given the 01*0 Turing Machine and a list of input strings. The figure below shows all the outputs that were printed to the screen, and after reading “endinput”, the program stopped and wrote the file “01*0-output.txt”

- 

```
jennifertran@jennifers-air-2 project3 % python3 traceTM-jtran4.py
Enter TM file name: example1.csv
Reading TM file...: 16it [00:00, 51622.20it/s]
Enter input string or endinput: 010
Enter max steps: 5

Name of the machine: 01*0
Initial input string: 010
Depth of the tree of configurations: 4.
Total transitions: 4.
String 010 accepted in 4 transitions.
,q0,0,10
x,q1,1,0
xy,q1,0,
xyx,q2,_,
xyx_,qacc,_,

Enter input string or endinput: 0101
Enter max steps: 5

Name of the machine: 01*0
Initial input string: 0101
Depth of the tree of configurations: 4.
Total transitions: 4.
String 0101 rejected in 4 transitions.

Enter input string or endinput: endinput
```

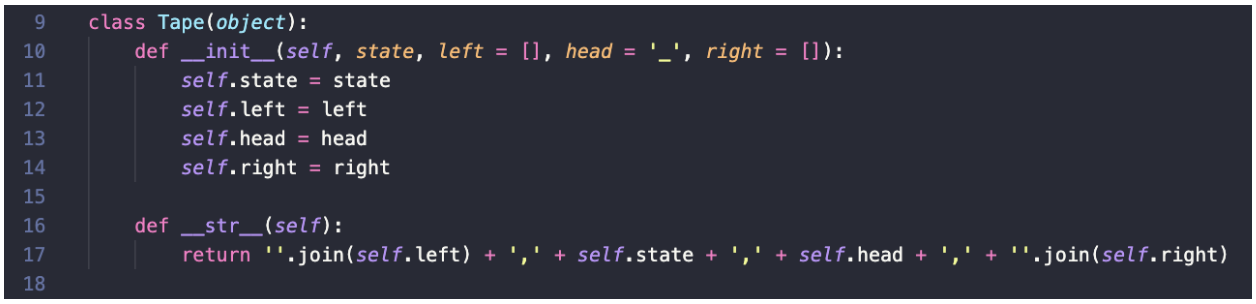
- Output format:
    - 

```
project3 > 01*0-output.txt
Name of the machine: 01*0.

Initial input string: 010
Depth of the tree of configurations: 4.
Total transitions: 4.
String 010 accepted in 4 transitions.
,q0,0,10
x,q1,1,0
xy,q1,0,
xyx,q2,_,
xyx_,qacc,_,

Initial input string: 0101
Depth of the tree of configurations: 4.
Total transitions: 4.
String 0101 rejected in 4 transitions.
```

5. The language you used, and a list of libraries you invoked.
- Language: Python3
- List of libraries:
    - csv
    - time
    - tdqm
    - sys
    - os
    - collections

6. A description of the key data structures you used, especially for the internal representation of wffs, assignments, and choice point stacks.

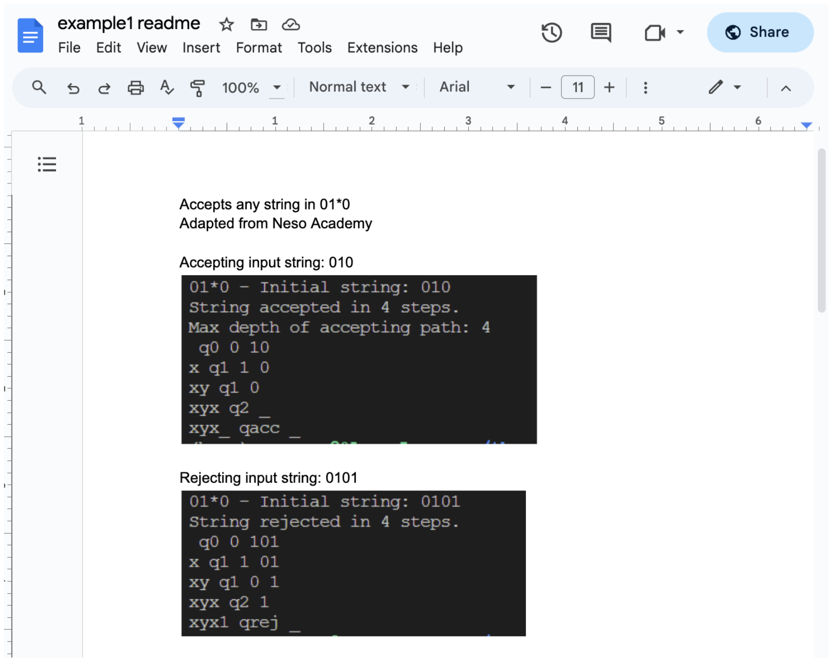
- I applied BFS to trace all the nondeterministic paths of the tree configurations from the given Turing Machine and the input string. I have dequeue to store the 3-tuple (tape, level, path).
- The variable tape is an object that stores the current state, the current tape head, and characters on the left and right of the tape head.
-



```python
class Tape(object):
    def __init__(self, state, left = [], head = '_', right = []):
        self.state = state
        self.left = left
        self.head = head
        self.right = right

    def __str__(self):
        return ''.join(self.left) + ',' + self.state + ',' + self.head + ',' + ''.join(self.right)

```

- The level is the level of the tree that the tape is at, while the path stores all transitions from the start state to the final state (which can be either accept state, reject state, or neither if reaching the limit steps).
- I used the BFS to get all the transitions at each level, and I used an unordered map (or dictionary in Python) to keep track of the depth of the tree.
- I started the queue with the initial tape at the start state, level 0, and a path that starts that includes the start state. Then I started looping through the queue with two conditions: the queue is not empty and the program has not reached the limit steps.
- In every loop cycle, I pop an element out of the queue which gives the 3-tuple (tape, level, path). If the level is new, I added it as the new key to the visited dictionary. I also append the tape to its current level in the visited dictionary for debugging purposes so that I can keep track of the tree of configurations. The program continues checking if the current state is an accept state. If so, it then halts and returns the output. If not, it continues searching for new transitions, appends them to the queue, and updates the level and path. If there is no transition, the program moves on to another element in the queue.

7. A discussion as to what test cases you added and why you decided to add them (what did they tell you about the correctness of your code). This discussion should include listings of the strings used for trace-nfa (if done) and input to regex2nfa (if done).

- example1.csv (Test case from the Drive)
    - This program produces the right output for the input strings following the rule 01*0.
    - I used the readme file provided by the student who created this testcase to test my program.
    - example1 readme:
        - 
          example1 readme
File Edit View Insert Format Tools Extensions Help
Share
100% Normal text Arial 11

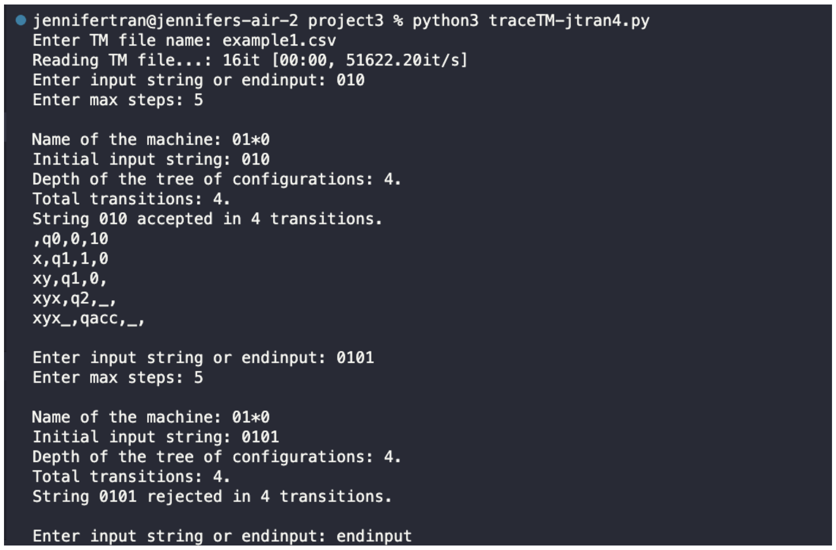
Accepts any string in 01*0
Adapted from Neso Academy

Accepting input string: 010

```
01*0 - Initial string: 010
String accepted in 4 steps.
Max depth of accepting path: 4
 q0 0 10
x q1 1 0
xy q1 0
xyx q2 _
xyx_ qacc _
```

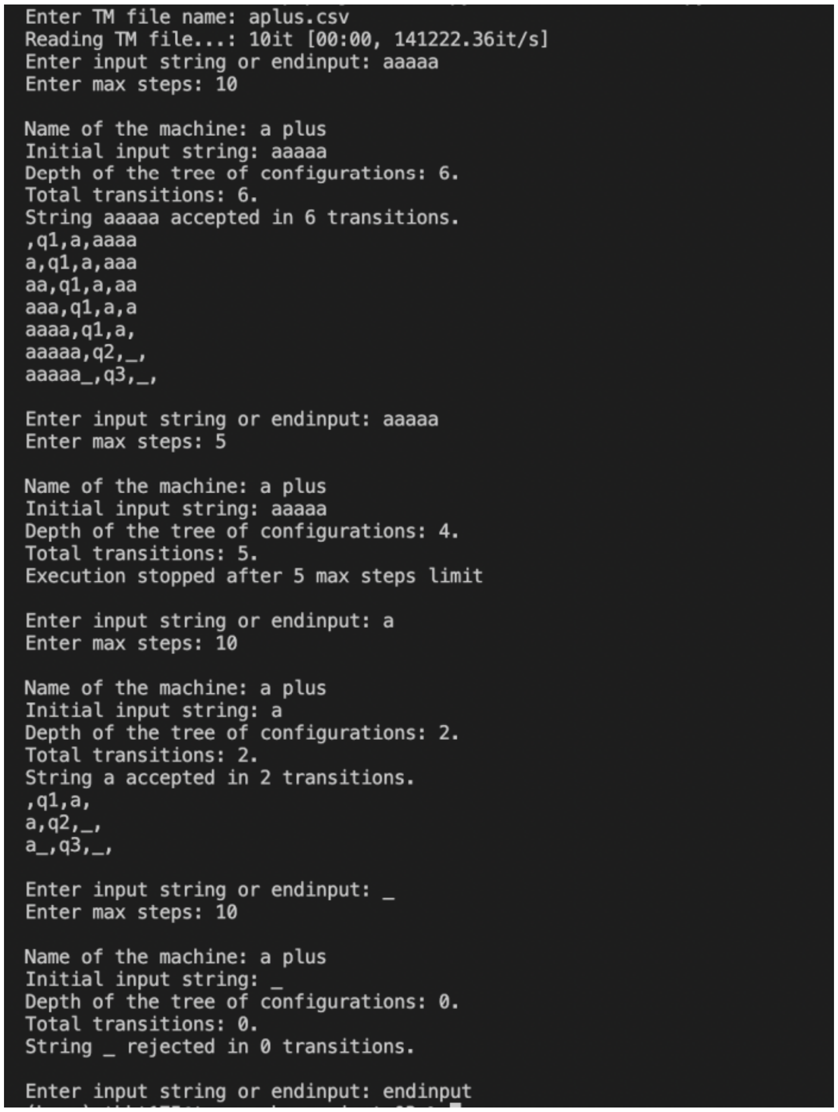
Rejecting input string: 0101

```
01*0 - Initial string: 0101
String rejected in 4 steps.
 q0 0 101
x q1 1 01
xy q1 0 1
xyx q2 1
xyx1 qrej _
```

    - My program's input/output
        - 

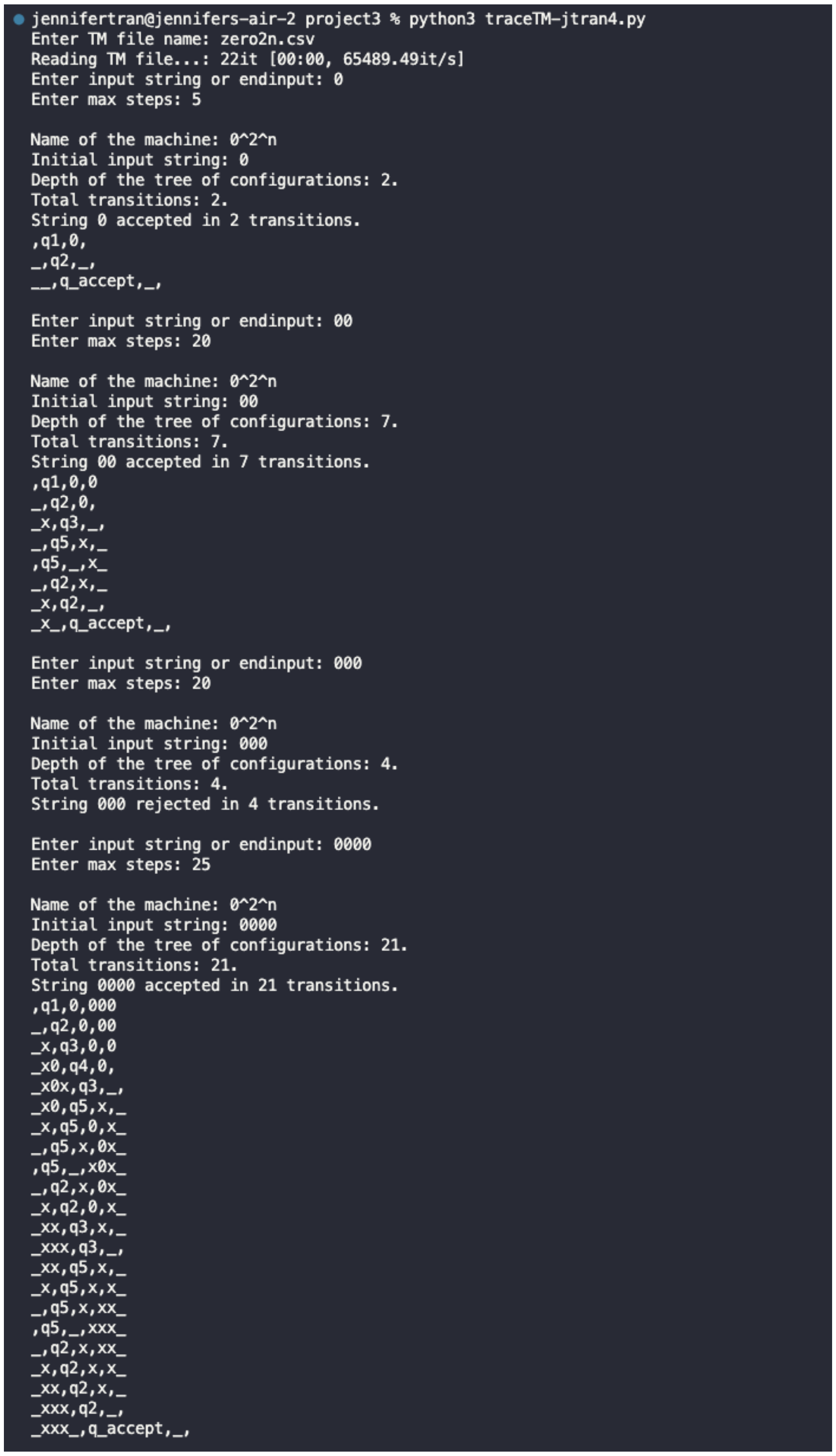
```
jennifertran@jennifers-air-2 project3 % python3 traceTM-jtran4.py
Enter TM file name: example1.csv
Reading TM file...: 16it [00:00, 51622.20it/s]
Enter input string or endinput: 010
Enter max steps: 5

Name of the machine: 01*0
Initial input string: 010
Depth of the tree of configurations: 4.
Total transitions: 4.
String 010 accepted in 4 transitions.
,q0,0,10
x,q1,1,0
xy,q1,0,
xyx,q2,_,
xyx_,qacc,_,

Enter input string or endinput: 0101
Enter max steps: 5

Name of the machine: 01*0
Initial input string: 0101
Depth of the tree of configurations: 4.
Total transitions: 4.
String 0101 rejected in 4 transitions.

Enter input string or endinput: endinput
```

        - The output file name is "01*0-output.txt"

- aplus.csv: (Test case from the Drive)
    - The Turing Machine accepts the language a+
    - The program outputs the correct answers for all the input strings.
    - 

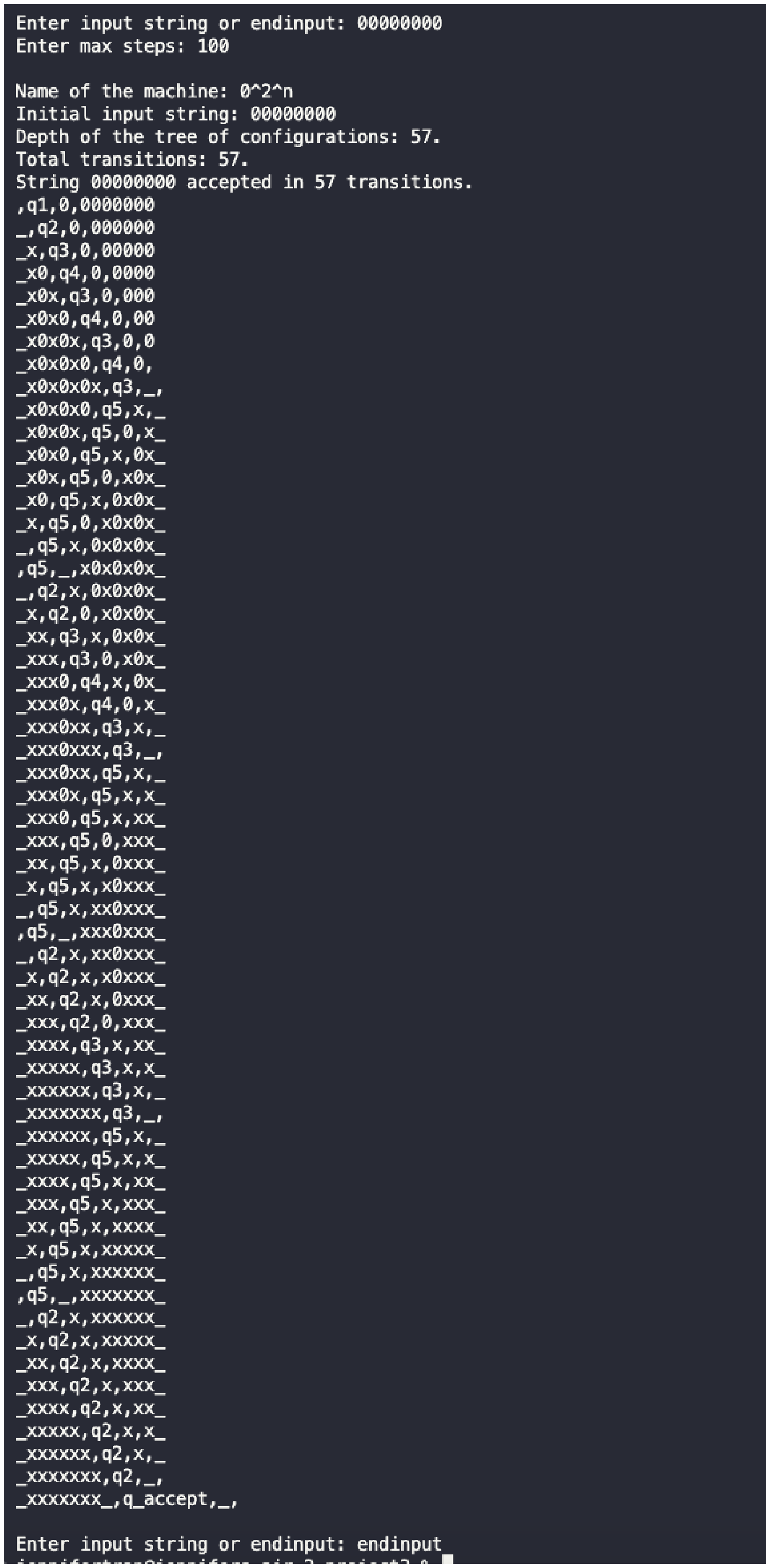
```
Enter TM file name: aplus.csv
Reading TM file...: 10it [00:00, 141222.36it/s]
Enter input string or endinput: aaaaa
Enter max steps: 10

Name of the machine: a plus
Initial input string: aaaaa
Depth of the tree of configurations: 6.
Total transitions: 6.
String aaaaa accepted in 6 transitions.
,q1,a,aaaa
a,q1,a,aaa
aa,q1,a,aa
aaa,q1,a,a
aaaa,q1,a,
aaaaa,q2,_,
aaaaa_,q3,_,

Enter input string or endinput: aaaaa
Enter max steps: 5

Name of the machine: a plus
Initial input string: aaaaa
Depth of the tree of configurations: 4.
Total transitions: 5.
Execution stopped after 5 max steps limit

Enter input string or endinput: a
Enter max steps: 10

Name of the machine: a plus
Initial input string: a
Depth of the tree of configurations: 2.
Total transitions: 2.
String a accepted in 2 transitions.
,q1,a,
a,q2,_,
a_,q3,_,

Enter input string or endinput: _
Enter max steps: 10

Name of the machine: a plus
Initial input string: _
Depth of the tree of configurations: 0.
Total transitions: 0.
String _ rejected in 0 transitions.

Enter input string or endinput: endinput
```

    - The output file name is “aplus-output.txt”

8. If you did any extra programs, or attempted any extra test cases, describe them separately.
    - zero2n.csv: (My test case, also in the drive)
        - The given Turing Machine accepts the language L = {0^(2^n) | n >= 0}. It means that the machine would accept 0, 00, 0000, 00000000, etc. and reject 000, 00000, 000000, etc.
        - The program produces the right output for test cases.
        - 

```
jennifertran@jennifers-air-2 project3 % python3 traceTM-jtran4.py
Enter TM file name: zero2n.csv
Reading TM file...: 22it [00:00, 65489.49it/s]
Enter input string or endinput: 0
Enter max steps: 5

Name of the machine: 0^2^n
Initial input string: 0
Depth of the tree of configurations: 2.
Total transitions: 2.
String 0 accepted in 2 transitions.
,q1,0,
_,q2,_,
__,q_accept,_,

Enter input string or endinput: 00
Enter max steps: 20

Name of the machine: 0^2^n
Initial input string: 00
Depth of the tree of configurations: 7.
Total transitions: 7.
String 00 accepted in 7 transitions.
,q1,0,0
_,q2,0,
_x,q3,_,
_,q5,x,_
,q5,_,x_
_,q2,x,_
_x,q2,_,
_x_,q_accept,_,

Enter input string or endinput: 000
Enter max steps: 20

Name of the machine: 0^2^n
Initial input string: 000
Depth of the tree of configurations: 4.
Total transitions: 4.
String 000 rejected in 4 transitions.

Enter input string or endinput: 0000
Enter max steps: 25

Name of the machine: 0^2^n
Initial input string: 0000
Depth of the tree of configurations: 21.
Total transitions: 21.
String 0000 accepted in 21 transitions.
,q1,0,000
_,q2,0,00
_x,q3,0,0
_x0,q4,0,
_x0x,q3,_,
_x0,q5,x,_
_x,q5,0,x_
_,q5,x,0x_
,q5,_,x0x_
_,q2,x,0x_
_x,q2,0,x_
_xx,q3,x,_
_xxx,q3,_,
_xx,q5,x,_
_x,q5,x,x_
_,q5,x,xx_
,q5,_,xxx_
_,q2,x,xx_
_x,q2,x,x_
_xx,q2,x,_
_xxx,q2,_,
_xxx_,q_accept,_,
```

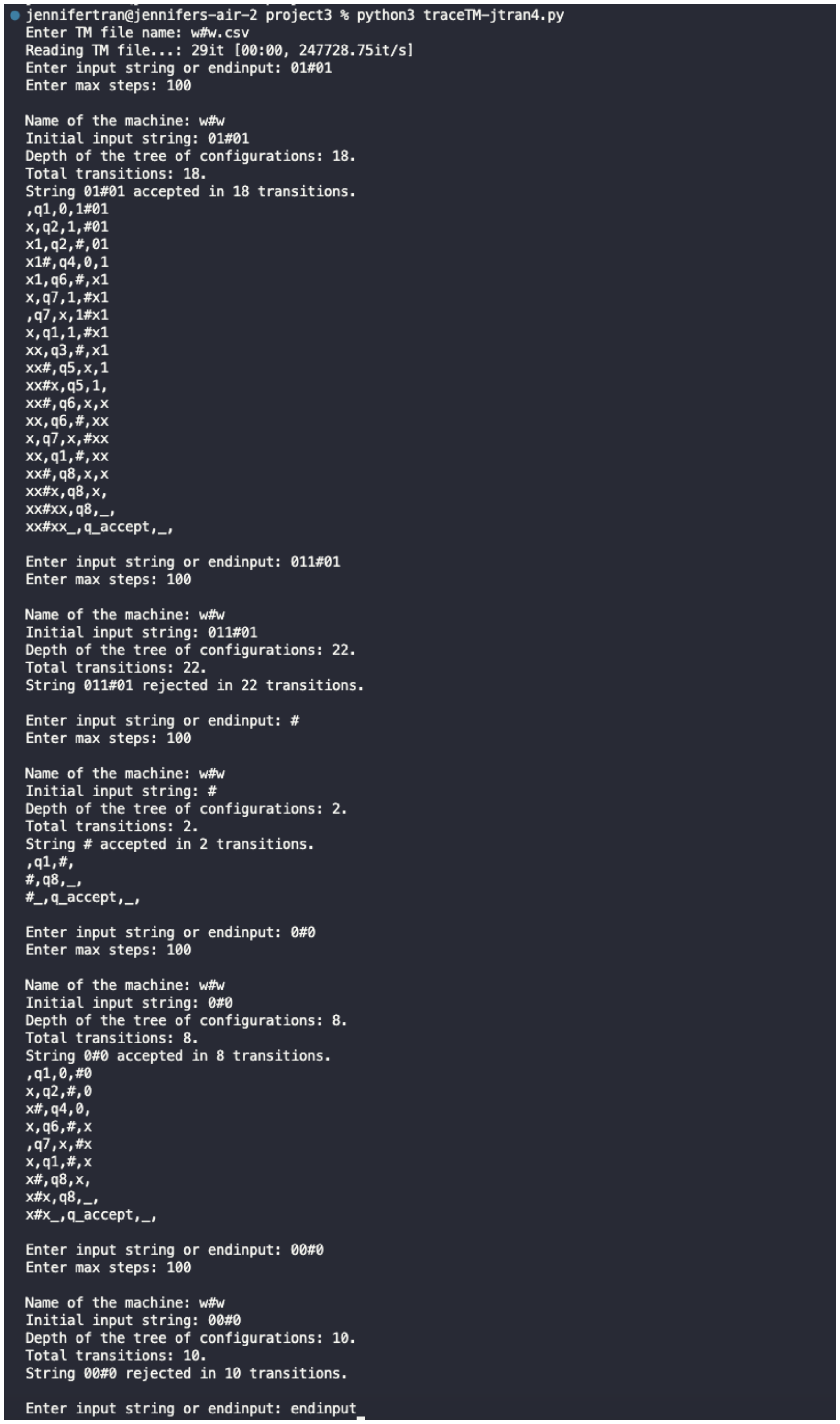
-



```
Enter input string or endinput: 00000000
Enter max steps: 100

Name of the machine: 0^2^n
Initial input string: 00000000
Depth of the tree of configurations: 57.
Total transitions: 57.
String 00000000 accepted in 57 transitions.
,q1,0,0000000
_,q2,0,000000
_x,q3,0,00000
_x0,q4,0,0000
_x0x,q3,0,000
_x0x0,q4,0,00
_x0x0x,q3,0,0
_x0x0x0,q4,0,
_x0x0x0x,q3,_,
_x0x0x0,q5,x,_
_x0x0x,q5,0,x_
_x0x0,q5,x,0x_
_x0x,q5,0,x0x_
_x0,q5,x,0x0x_
_x,q5,0,x0x0x_
_,q5,x,0x0x0x_
,q5,_,x0x0x0x_
_,q2,x,0x0x0x_
_x,q2,0,x0x0x_
_xx,q3,x,0x0x_
_xxx,q3,0,x0x_
_xxx0,q4,x,0x_
_xxx0x,q4,0,x_
_xxx0xx,q3,x,_
_xxx0xxx,q3,_,
_xxx0xx,q5,x,_
_xxx0x,q5,x,x_
_xxx0,q5,x,xx_
_xxx,q5,0,xxx_
_xx,q5,x,0xxx_
_x,q5,x,x0xxx_
_,q5,x,xx0xxx_
,q5,_,xxx0xxx_
_,q2,x,xx0xxx_
_x,q2,x,x0xxx_
_xx,q2,x,0xxx_
_xxx,q2,0,xxx_
_xxxx,q3,x,xx_
_xxxxx,q3,x,x_
_xxxxxx,q3,x,_
_xxxxxxx,q3,_,
_xxxxxx,q5,x,_
_xxxxx,q5,x,x_
_xxxx,q5,x,xx_
_xxx,q5,x,xxx_
_xx,q5,x,xxxx_
_x,q5,x,xxxxx_
_,q5,x,xxxxxx_
,q5,_,xxxxxxx_
_,q2,x,xxxxxx_
_x,q2,x,xxxxx_
_xx,q2,x,xxxx_
_xxx,q2,x,xxx_
_xxxx,q2,x,xx_
_xxxxx,q2,x,x_
_xxxxxx,q2,x,_
_xxxxxxx,q2,_,
_xxxxxxx_,q_accept,_,

Enter input string or endinput: endinput
```

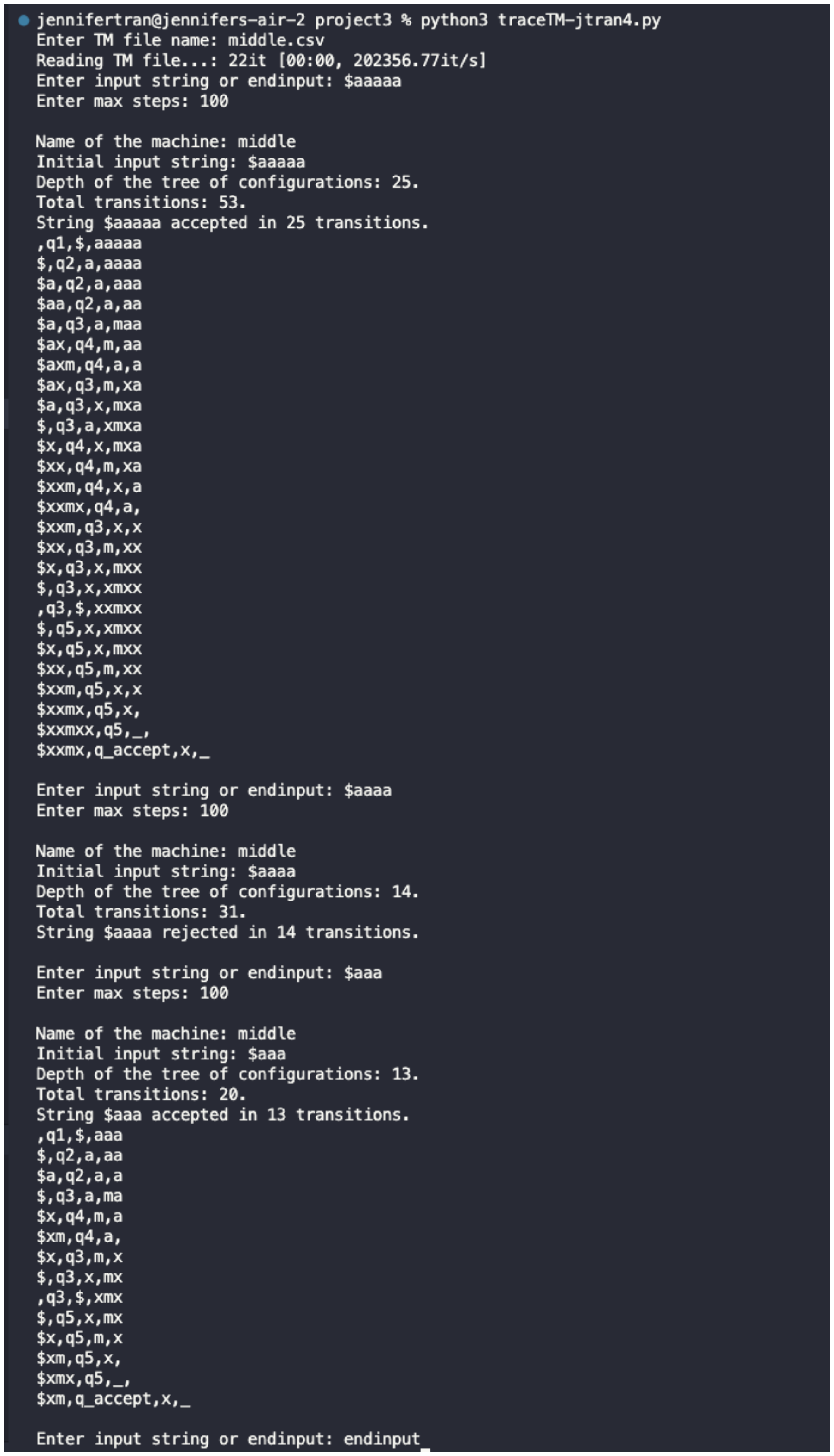
- The output file name is “0^2^n-output.txt”

- <u>w#w.csv: (My test case, also in the Drive)</u>
    - The Turing Machine accepts the language L = {w#w | w ∈ {0,1}*}. It accepts the strings such as #, 01#01, 000#000, 101#101, etc., and rejects 011#01, #000, etc.
    - The program produces the right output for test cases.



```
jennifertran@jennifers-air-2 project3 % python3 traceTM-jtran4.py
Enter TM file name: w#w.csv
Reading TM file...: 29it [00:00, 247728.75it/s]
Enter input string or endinput: 01#01
Enter max steps: 100

Name of the machine: w#w
Initial input string: 01#01
Depth of the tree of configurations: 18.
Total transitions: 18.
String 01#01 accepted in 18 transitions.
,q1,0,1#01
x,q2,1,#01
x1,q2,#,01
x1#,q4,0,1
x1,q6,#,x1
x,q7,1,#x1
,q7,x,1#x1
x,q1,1,#x1
xx,q3,#,x1
xx#,q5,x,1
xx#x,q5,1,
xx#,q6,x,x
xx,q6,#,xx
x,q7,x,#xx
xx,q1,#,xx
xx#,q8,x,x
xx#x,q8,x,
xx#xx,q8,_,
xx#xx_,q_accept,_,

Enter input string or endinput: 011#01
Enter max steps: 100

Name of the machine: w#w
Initial input string: 011#01
Depth of the tree of configurations: 22.
Total transitions: 22.
String 011#01 rejected in 22 transitions.

Enter input string or endinput: #
Enter max steps: 100

Name of the machine: w#w
Initial input string: #
Depth of the tree of configurations: 2.
Total transitions: 2.
String # accepted in 2 transitions.
,q1,#,
#,q8,_,
#_,q_accept,_,

Enter input string or endinput: 0#0
Enter max steps: 100

Name of the machine: w#w
Initial input string: 0#0
Depth of the tree of configurations: 8.
Total transitions: 8.
String 0#0 accepted in 8 transitions.
,q1,0,#0
x,q2,#,0
x#,q4,0,
x,q6,#,x
,q7,x,#x
x,q1,#,x
x#,q8,x,
x#x,q8,_,
x#x_,q_accept,_,

Enter input string or endinput: 00#0
Enter max steps: 100

Name of the machine: w#w
Initial input string: 00#0
Depth of the tree of configurations: 10.
Total transitions: 10.
String 00#0 rejected in 10 transitions.

Enter input string or endinput: endinput
```

    -
    - The output file name is “w#w-output.txt”

- middle.csv
    - This TM will accept if the middle symbol is ‘a’.
    - The $ is just the start of the stack
    - 

```
● jennifertran@jennifers-air-2 project3 % python3 traceTM-jtran4.py
  Enter TM file name: middle.csv
  Reading TM file...: 22it [00:00, 202356.77it/s]
  Enter input string or endinput: $aaaaa
  Enter max steps: 100

  Name of the machine: middle
  Initial input string: $aaaaa
  Depth of the tree of configurations: 25.
  Total transitions: 53.
  String $aaaaa accepted in 25 transitions.
  ,q1,$,aaaaa
  $,q2,a,aaaa
  $a,q2,a,aaa
  $aa,q2,a,aa
  $a,q3,a,maa
  $ax,q4,m,aa
  $axm,q4,a,a
  $ax,q3,m,xa
  $a,q3,x,mxa
  $,q3,a,xmxa
  $x,q4,x,mxa
  $xx,q4,m,xa
  $xxm,q4,x,a
  $xxmx,q4,a,
  $xxm,q3,x,x
  $xx,q3,m,xx
  $x,q3,x,mxx
  $,q3,x,xmxx
  ,q3,$,xxmxx
  $,q5,x,xmxx
  $x,q5,x,mxx
  $xx,q5,m,xx
  $xxm,q5,x,x
  $xxmx,q5,x,
  $xxmxx,q5,_,
  $xxmx,q_accept,x,_

  Enter input string or endinput: $aaaa
  Enter max steps: 100

  Name of the machine: middle
  Initial input string: $aaaa
  Depth of the tree of configurations: 14.
  Total transitions: 31.
  String $aaaa rejected in 14 transitions.

  Enter input string or endinput: $aaa
  Enter max steps: 100

  Name of the machine: middle
  Initial input string: $aaa
  Depth of the tree of configurations: 13.
  Total transitions: 20.
  String $aaa accepted in 13 transitions.
  ,q1,$,aaa
  $,q2,a,aa
  $a,q2,a,a
  $,q3,a,ma
  $x,q4,m,a
  $xm,q4,a,
  $x,q3,m,x
  $,q3,x,mx
  ,q3,$,xmx
  $,q5,x,mx
  $x,q5,m,x
  $xm,q5,x,
  $xmx,q5,_,
  $xm,q_accept,x,_

  Enter input string or endinput: endinput
```

    - The output file name is “middle-output.txt”