This memo describes each step of the process to finish the assignment. It indicates the language of the implemented FA and the regular expression corresponding to it.

Python 3.6 was used to do this programming assignment and the code was tested on a Linux environment like UBUNTU.

Code Details:
The code has 2 parts. For both parts, it takes in a string as input. It raises an exception if the number is less than 0 or greater than 9, letters, and symbols. The Finite State Machine FSM resets the program in case of an exception.

PART# 1

Variables used:
Code = “02963” - Student ID
Lock_Status = True - Current status
State = 0 - FSM - current status

Functions used:
Lock(key, Status_Lock)

- “key” is an integer and it locks and unlocks the device if the following value is 1 or 4 respectively.
- “Status_Lock” represents whether the lock is locked or unlocked. It returns the status of the lock.

Digit(number, n)

- “Number” is an integer and it is the code of the lock
- “n” is also an integer and it is the current state of the finite state machine. It returns the value of the code based on the current state.

PART# 2
Variables used:
secs = 0
Rand_Code = random.randint(0,99999)
Lock_Status = True
State = numDigits(Rand_Code) - 1
Cur_State = State

Functions:
Lock(key, Status_Lock)

- “key” is an integer and it locks and unlocks the device if the following value is 1 or 4 respectively.
- “Status_Lock” represents whether the lock is locked or unlocked. It returns the status of the lock.

Digit(number, n)

- “Number” is an integer and it is the code of the lock
- “n” is also an integer and it is the current state of the finite state machine. It returns the value of the code based on the current state.

numDigits(num)

- “num” is an integer and it is the access code. It return the count which is the length of numbers used to calculate the number of states.

Check_Door(door)

- “door” is a boolean and it tells whether the door is locked or unlocked. It calculates the number of seconds it takes to crack the code. It returns a boolean value.

The time it takes to crack the code:

Average: 13650099 seconds