assumptions:
1. in the output, the order of display of states,letters,t_func,start and final can be in a random order(meaning that final states could be displayed 1st, then all the transitions,then the start state or any other such permutation)
2. the input.json file is in the same directory as the script.py

code explanation:
1. the os.path.exists checks if there is an output.json present, if it does os.remove deletes the file to prevent overwriting
2. the input.json is loaded to the dictionary nf at line 9
3. states1 stores the 2^n states in decimal, states2 contains the same in binary,power_state contains the power set of all states of nfa and the list y2 used in its creation, basically stores 1 subset each time , the subset stored in y2 keeps changing via the for loop, if the nth bit is 1 you include the nth no. in your subset,otherwise not
4. t_func is list of all transitions in the dfa,func is a single transition of dfa,out_state is output of a single transition of dfa
5. while iterating in the list of power sets of nfa states(or the list of dfa states), you choose an alphabet, for that alphabet you choose a transition from the nfa transition list and then you choose an element from the subset chosen, if for a given letter the letter and the input state from transtion of nfa are the same as the letter in the loop and the element of the subset, we add it to out_state, we keep doing this till all transitions are done for each element in the chosen subset for a given letter, then we append the subset,the letter and out_state to func and append func to t_func, then we clear both the out_state and func for the next letter(of the for loop), once each letter is iterated over for a given subset, the same process repeats for other subsets, giving us the t_func(set of all transitions of the dfa)
6. iterating over each element of each subset of the set of states of the nfa, if any element in any given subset is the same as an element in the final state of the nfa, we include that subset in the final state (dfa_fin) of the dfa, this happens for each subset of the nf states
7. we make a dictionary res made of states,letters,t_func,start,final
8. we dump this dictionary into a file named output.json using json.dump