NFA Algorithm:

1. Assuming No symbols or brackets For now.

1. Define a bool indicates whether we should insert empty state → is Epsilon
2. Define a list which contains our states → [] empty.
3. initialize characters idx which points to the current character of the given regular expression with -1
4. create initial state S1 (init=T, term= F, trans = {})
5. Append it to the states list
6. iterate For each state in the states list
   - 6.1. apply logic (state, char idx, is Epsilon, regex, states
   - 6.2. is Epsilon ~ is Epsilon → toggle it → initial
   - 6.3. char idx += ! is Epsilon we move the pointer in case that we are not @ Epsilon state

// we should continue the NFA algo, but lets stop here now & implement apply logic themselves

---

apply logic (state, char idx, is Epsilon, regex, states

1. create new empty state → Si
2. Append it to the states list → states.append[Si]
3. if (is Epsilon)
   - 3.1. state.Trans {'ε'}.append [Si]
4. else
   - 4.1. if (regex[char idx + 1] is alphanumeric)
     - 4.1.1. state.Trans[*].append [Si]
   - 4.2. else

// we should think how to handle these cases.

abcDE: (9)*