S = 'abC_pqr'

for this discussion treat this as space than underscore.

---

① Go through the string in character by character fashion. And at any given moment you must know whether you're inside the word or outside the word.

IN = 1 .... inside the word

OUT = 2 ... outside the word

currentState = OUT

w_cnt = 0

We've visiting string character by character.

Do the following activity on each character.

① [currentState == OUT

currentCharacter == Non-white space]

↘ Marks the start of new word.

currentState ← IN

w_cnt ← w_cnt + 1.

② [curvenState == OUT.

currentChavacter == White space]

↘ Do nothing.

③ [currentState == IN

currentChavacter == Non-white space]

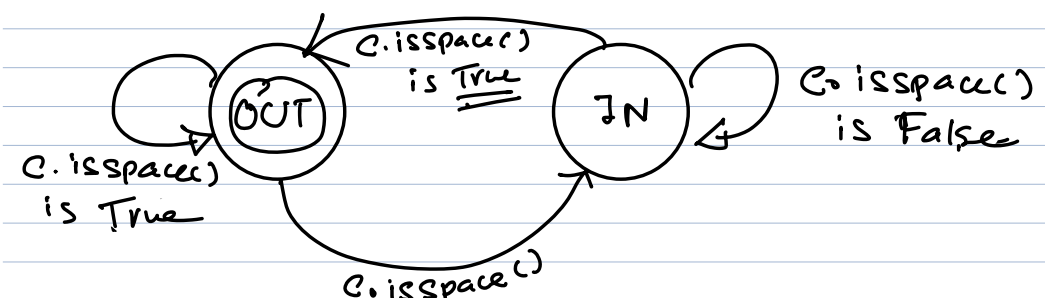
↘ Do nothing.

④ [current State == IN.

currentChavacter == white space]

↘ current State == OUT

```
for c in s:
    if currentState is OUT and
        not c.isspace() :
        current State = IN
        w_cnt = w_cnt + 1
    elif current State is IN and
        c. isspace() is True :
        current State = OUT
```



[Action:
currentState = IN]

↳ Not for beginner's.

D.F.A. = $(Q, \Sigma, \delta : Q \times \Sigma \to Q, q_0, Q_F)$

Q = Non-empty set of steps.

Σ = Non-empty set of alphabets.

δ = State Transition Function

q_0 = $q_0 \in Q$, initial state

Q_F = $Q_F \neq \phi$ and $Q_F \subseteq Q$.

Set of accepting states.

↳ Not for beginners & faint hearted

S = ' __ abc \t \t _ pqrs __ t __ \n \t '

c c c c c c c c c c c c c c c c c c c c

cS = OUT → IN → OUT → IN → OUT → IN → OUT

w_cnt = 0 → 1 → 2 → 3