Q3 What is predictive user? Explain its block diagram with an implementation example showing its working.

Answer ⇒ A predictive parser a recursive descent parser with no backup. It is a top-down parser that does not require back-tracking. At each step, the choice of the rule to be expanded is made upon the next terminal symbol.

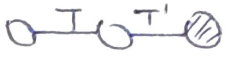
→ Consider the following grammer–

$E \to E+T \mid T$, $T \to T*F \mid F$, $F \to (E) \mid id$

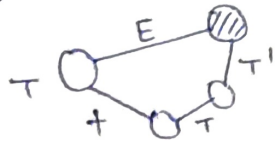
→ After removing left recursion.

$E \to TT'$, $T' \to +TT' \mid E$, $T \to FT''$, $T'' \to *FT'' \mid E$, $F \to (E)' \mid id$

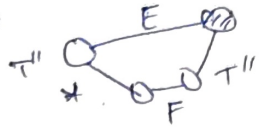
* **Step 1:** – Make a transition diagram (DFA/NFA) for every grammer.

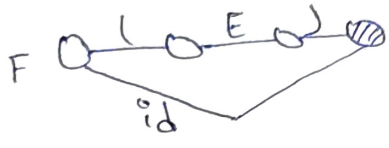
- $E \to TT'$ ⇒ 

- $T' \to +TT' \mid \epsilon$ ⇒



- $T \to FT''$ ⇒ 

- $T'' \to *FT'' \mid \epsilon$ ⇒ 

- $F \to (E) \mid id$ ⇒ 

* **Step 2:** → Optimise the DFA by decreasing the no. of states yielding the final transition diagram.

* **Step 3:** → Stimulation in the input string

- Start from the starting state. If a terminal arrives consume it and move to next state.
- If a non-terminal arrived go to the state of the DFA of non-terminal & return on reached up to the final state.
- Return to actual DFA and keep doing parsing.
- If one complete reading the input string completely, you reach a final state and string is successfully parsed.