Experiment No:6

AIM: To implement a program to find follow() for any given Production.
THEORY:Follow(X) to be the set of terminals that can appear immediately to the right of Non-Terminal X in some sentential form.

Algorithm:
Computation of FOLLOW

Follow (A) is defined as the collection of terminal symbols that occur directly to the right of A.

$$FOLLOW(A) = \{a \mid S \Rightarrow^* \alpha A a \beta \text{ where } \alpha, \beta \text{ can be any strings}\}$$

If S is the start symbol, FOLLOW (S) ={$}
If production is of form $A \rightarrow \alpha B \beta$, $\beta \neq \varepsilon$.

(a) If FIRST (β) does not contain ε then, FOLLOW (B) = {FIRST (β)}

Or

(b) If FIRST (β) contains ε (i. e. , $\beta \Rightarrow^* \varepsilon$), then

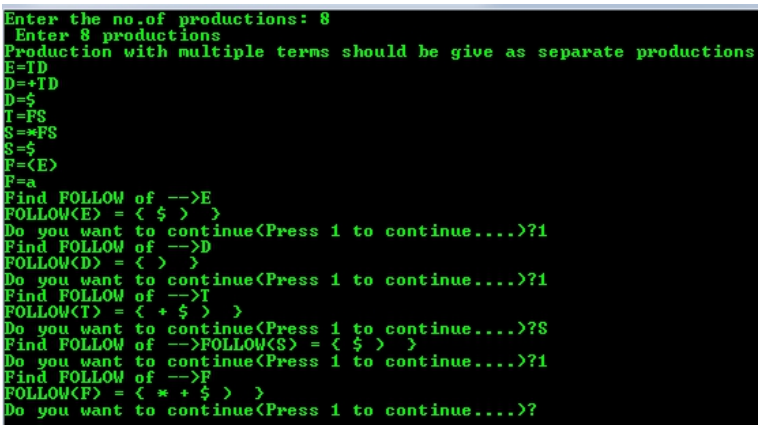
FOLLOW (B) = FIRST (β) – {ε} ∪ FOLLOW (A)

∵ when β derives ε, then terminal after A will follow B.

If production is of form $A \rightarrow \alpha B$, then Follow (B) ={FOLLOW (A)}.

COMPUTING ENVRONMENT
Platform: ubuntu
Programming Language: C / C++ / Java

Expacted Output:



Conclusion: Thus the program to find FOLLOW() is implemented.

Viva Voce Questions:

1. What is Follow()**?**

Answer: **Follow (A) is defined as the collection of terminal symbols that occur directly to the right of A.**

2. Why FOLLOW() is calculated**?**

Answer:FOLLOW can make a Non-terminal vanish out if needed to generate the string from the parse tree.FOLLOW sets for a given grammar so that the parser can properly apply the needed rule at the correct position.