Consider the right recursive grammar E ☛T + E | T T ☛V * T | V V ☛<id>	Top-down Parsing with Back tracking
Left Factoring Common prefix requires applying left factoring E ☞T + E | T T ☞V * T | V Left factor E E ☞T + E | T E ☞ TE” E” ☞ + E |ε Left factor T T ☞ V * T | V T☞ VT” T”☞ * T | ε	Top-down parsing without back tracking

Modified Left Factored Grammar
 	
E ::= T { +T}*
T ::= V{* V}*
V ::= <id>
	RD parser

Consider the Left Recursive Grammar E☛E+T|T T☛T*V|V V☛<id>	Not Used as it is for Top-Down Parsing
Eliminating left-recursion Apply left recursion elimination E ☞E+T|T T☞T*V|V Eliminate left recursion of E E☞TE’ E’☞ ε | + TE’ Eliminating left recursion of T T☞VT’ T’☞ ε | + VT’	LL1 / Table driven / Top-down Parsing
Operator Grammar using (OPM)	Operator Precedence Parser (Bottom Up Parser)

	+	*	$
+	·>	<·	·>
*	·>	·>	·>
$	<·	<·	·>