Manil S. Malla

wcrecent@gmail.com

To convert Infix expression into postfix expression using predictive parser based on grammar.

Compiler Lab 1

Infix to Postfix expression.

* **Introduction**
* Syntax Directed Definition
* Syntax Directed Translation
* Predictive Parsing
* Left Recursive Grammer
* **Objective**
* Translate an infix expression into postfix expression using predictive parser based on grammar.

Expr -> term R

R -> + term {print(‘+’)} R

R -> - term {print(‘-’)} R

Rest -> E

Term -> 0 {print(‘0’)}

Term -> 1 {print(‘1’)}

.

.

.

Term -> 9 {print(‘9’)}

* **Code**

*#include<stdio.h>*

*#include<conio.h>*

*#include<ctype.h>*

*int lookahead;*

*int main()*

*{*

*lookahead = getchar();*

*expr();*

*putchar('\n');*

*getch();*

*return 0;*

*}*

*void expr()*

*{*

*term();*

*while(1)*

*{*

*if(lookahead == '+')*

*{*

*match('+');*

*term();*

*putchar('+');*

*}*

*else if(lookahead == '-')*

*{*

*match('-');*

*term();*

*putchar('-');*

*}*

*else*

*{*

*break;*

*}*

*}*

*}*

*void term()*

*{*

*if(isdigit(lookahead))*

*{*

*putchar(lookahead);*

*match(lookahead);*

*}*

*else*

*{*

*error();*

*}*

*}*

*void match()*

*{*

*int t;*

*if(lookahead == t)*

*{*

*lookahead = getchar();*

*}*

*else*

*{*

*error();*

*}*

*}*

*void error()*

*{*

*printf("\n\n\t\t\tSyntax Error!!!");*

*exit(1);*

*}*

* Output

Input = 9+5-2

Output = 95+2-