PES UNIVERSITY

Name	SRN
C Diya	PES1201700246
Namrata R	PES1201700921
Chiranth J	PES1201701438

COMPILER DESIGN MINI PROJECT

Language : C

Constructs: While, Structures, Switch Case

Regex

- `Letter = [a-zA-Z]`
- `Digit = [0-9]`
- `Id = (letter| _ )(letter|_|digit)*`
- `Num = digit+`
- `Character = '.'`
- `Label = Number | 'letter' | "letter*"`
- `Number = digit+(.digit)?([Ee][+-]?digit+)?`

Grammar

- `Program → (Prepoc_directive)* (Structure)* int main(){Statement} | (Prepoc_directive)* (Structure)* void main(){Statement}`
- `dataType → int | float | struct id | struct | char`
- `Prepoc_directive → (#include<directive>)`
- `Directive → stdio.h| string.h| conio.h| math.h`
- `Structure →  typedef struct id{ Structinterals}id; | struct id{ Structinterals};`

- Structinterals → Declaration;Structinterals | Declaration;
- Statement → break; | return 0; | declaration;Statement | assignment;Statement | while_stat Statement | switch_stat Statement | ; | epsilon
- Idloop → .idIDloop|.id|.id[num]| epsilon
- Declaration → dataType Dectype
- Dectype → decexp, Dectype | decexp
- Decexp → decid|decid=expression
- Decid → idIDloop | idIDloop[num]
- Assignment -> idIDloop = expression |id[num]IDloop = expression | idIDloop = { Expression_set (Expression)}| id[num]IDloop = { Expression_set (Expression)}
- Expression_set → Expression,Expression_set|epsilon
- while_stat -> while(condition){statement} | while(condition) statement
- condition-> relationalExpression | logicalExpression | idIDloop | Number | Character | id[num]IDloop
- unaryOperator -> ++ | --
- relationalOperator -> == | >= | <= | > | < | !=
- logicalOperator -> && | \|\| |!
- Expression -> Expression + T | Expression -T | T
- T -> T * F | T/ F | T % F | F
- F -> id[num]IDloop unaryOperator |idIDloop unaryOperator | unaryOperator idIDloop | unaryOperator id[num]IDloop | M
- M-> idIDloop | id[num]IDloop | Number | Character | (Expression)
- relationalExpression -> Expression relationalOperator Expression
- logicalExpression -> Logical_condition relationalExpression
- Logical_condition -> (relationalExpression logicalOperator) Logical_condition| (relationalExpression logicalOperator)
- switch_stat -> switch(condition){ Cases Defaultstat}
- Cases → (case label:statement)Cases|(case label:statement)
- Defaultstat → (default: statement)| epsilon

Token Table

Pattern	Lexeme	Token Name	Token
int | char | float | if | while | struct | typedef | return | break | default | case | switch | void	int, char, float, if, while, struct, typedef, return, break, default, case, switch, void	<keyword,key>	<keyword,switch>
[a-zA-Z_][a-zA-Z0-9_]*	-	id	<id,1>
[0-9]+(.[0-9)?([Ee][+-]?[0-9]+)?[0-9]*	-	num	<num,5>
> | < | >= | <= | == | !=	-	relop	<relop, ==>
&& | | | !	-	logop	<logop, &&>
-- , ++	-	unaryop	<unaryop,++>
+ | - | * | / | %	-	arithop	< arithop , - >
=	-	assign	<assign,=>
, | ; | ’ | ” | { | } | [|] | (|)	-	-	yytext
\/\/.* | \/*([a-zA-Z]*|[0-9]*|[(){};,"']|[\n|\t|' '] |\/\/.*|(\/*))**\/	;	-	-
[\n | \t | ’ ‘]	;	-	-