Course Code and Name: 2CS701 Compiler Construction

Name and Roll number: 20BCE297 Tilak Vasu

Practical 7: - To implement grammar rules for control statements, and Loop control

Code:

.l file

%{

#*include*<stdio.h>

#*include* "y.tab.h"

extern int yylval;

%}

%%

"for" {*return* fortoken;}

"while" {*return* whiletoken;}

"int" {*return* datatype;}

"=" {*return* assignment;}

";" {*return* semicolon;}

" " {*return* space;}

[0-9]+ {yylval=atoi(yytext); *return* digit;}

"++"|"--" {*return* unary;}

"(" {*return* openingbracket;}

")" {*return* closingbracket;}

[a-zA-Z][a-zA-Z0-9]\* { *return* identifier;}

">"|"<"|"=="|"<="|">=" {*return* condition;}

"{" {*return* curlyopening;}

"}" {*return* curlyclosing;}

.  {*return* codelogic;}

%%

int yywrap()

{

*return* 1;

}

.y file

%{

#include<stdio.h>

%}

%token whiletoken fortoken space datatype assignment semicolon unary openingbracket closingbracket identifier condition curlyopening curlyclosing codelogic digit

%%

S: fortoken M curlyopening curlyclosing {printf("For loop derived!"); return 0;}

| whiletoken N curlyopening curlyclosing{printf("While loop derived!"); return 0;};

M: openingbracket datatype space identifier assignment digit semicolon identifier condition digit semicolon identifier unary closingbracket {}

| openingbracket identifier assignment digit semicolon identifier condition digit semicolon identifier unary closingbracket {};

N: openingbracket datatype identifier condition digit closingbracket {}

| openingbracket identifier condition digit closingbracket{};

%%

void yyerror()

{

printf("For not derived!");

}

void main()

{

yyparse();

}

Output:

