33304

Assignment - 12

* Title :- LEX and YACC program to generate intermediate code.

* Problem Statement :- Write a program for intermediate code generation using LEX and YACC for control flow Statement (either while or ~~for~~ switch loop).

* Objectives :- 1. To understand 4th phase of compiler - intermediate code generation.
  2. To learn and use compiler writing tools
  3. To learn how to write 3 address code for a given statement.

* Theory :-

• Introduction →
In the analysis - synthesis phase of a compiler, the front end analyzes a source program and creates an intermediate representation, from which the backend generates target code. Ideally details of the source language are confided to the front end, and details of the target machine to the back end. The front end translates a source program into an intermediate representation from which the back end generates target code.

• Intermediate languages →
Three ways of intermediate representation
1. Syntax tree
2. Postfix Notation
3. Three address code

• Steps to execute the program →

33304

```
1. $ lex filename.l
2. $ yacc -d filename.y
3. $ cc lex.yy.c y.tab.c -ll -ly -lm
4. $ ./a.out
5. $ (eg: comp.l)
6. $ (eg: comp.y)
```

- Algorithm →

Write a LEX and YACC program to generate intermediate code for arithmetic expression LEX program.

1. Declaration of header files specially y.tab.h which contains declaration for letter, digit, expression.
2. End declaration section by %%
3. Match regular expression
4. If match found then convert it into char and store it in yylval.p where p is pointer declared in yacc.
5. Return token
6. If input contains new line character (\n) then return 0.
7. If input contains "." then return yytext [0].
8. End rule section action by %%
9. Declare main function
10. ⓐ Open file given at command line
11. ⓑ If any error occurs then print error and exit
12. ⓒ Assign file pointer fp to yyin
13. ⓓ call function yylex until file ends
14. End

YACC Program :

1. Declaration of header files.
2. Declare structure for three address code representation having fields of argument 1, argument 2, operator, result.

DATE __/__/____
PAGE NO ________

33304

3. Declare pointer of char type in union
4. Declare token expr of type p.
5. Give precedence to "*", "/"
6. Give precedence to "+", "-"
7. End of declaration statement by %%.
8. If final expression evaluates then add it to the table of three address code.
9. If input type is expr of the form
10. ⓐ exp "+" exp then add to table the argument 1, argument 2, operator
11. ⓑ exp "-" exp then add to table the argument 1, argument 2, operator
12. ⓒ exp "*" exp then add to table the argument 1, argument 2, operator.
13. ⓓ exp "/" exp then add to table the argument 1, argument 2, operator
14. ⓔ "(" exp ")" then assign $2 to $$.
15. ⓕ Digit or letter then assign $1 to $$.
16. End the section by %%.
17. Declare file *yyin externally
18. Declare main function and call yyparse function until yyin ends.
19. Declare yyerror for if any error occurs
20. Declare char pointer s to print error
21. Print error message
22. End of the program

* Conclusion :- Thus I have learnt about LEX and YACC and implemented a program for Intermediate code generation using LEX and YACC for control flow.