COMPILER LABORATORY
CS7611

LAB PROJECT

DONE BY :-
ANIKAITH ANAND (2017103076)
ARUN RAMANA B (2017103078)

Problem statement

Given an algorithm (pseudo code), construct a compiler that produces the corresponding C code.

Brief description

There is a lex and a yacc program (codeGen.l and codeGen.y respectively). The lex program is used to return keywords such as the beginning of a function, declarations and initialisations, loop statements, etc. The yacc program has the grammar productions required to produce the desired output.

The input is a text file that contains the pseudo code. It is opened in “read” mode in the yacc program. The output is the corresponding C code that is printed in the terminal.

Files in the project

codeGen.l
codeGen.y
eg.txt

TOOLS USED TO DEVELOP CODE

Operating system: OSX

Lex tool:

Lex is a computer program that generates lexical analyzers ("scanners" or "lexers").

Lex is commonly used with the yacc parser generator. Lex, originally written by Mike Lesk and Eric Schmidt and described in 1975, is the standard lexical analyzer generator on many Unix systems, and an equivalent tool is specified as part of the POSIX standard.

Lex reads an input stream specifying the lexical analyzer and outputs source code implementing the lexer in the C programming language. In addition to C, some old versions of Lex could also generate a lexer in Ratfor.

Yacc Tool: Yacc (Yet Another Compiler-Compiler)

It is a computer program for the Unix operating system developed by Stephen C. Johnson. It is a Look Ahead Left-to-Right (LALR) parser generator, generating a parser, the part of a compiler that tries to make syntactic sense of the source code, specifically a LALR parser, based on an analytic grammar written in a notation similar to Backus–Naur Form (BNF).Yacc is supplied as a standard utility on BSD and AT&T Unix. GNU-based Linux distributions include Bison, a forward-compatible Yacc replacement.

Steps to compile

1.Compile the lex code (codeGen.l) using the command:

lex codeGen.l

2.Compile the yacc code (codeGen.y) Using the command:

yacc codeGen.y

You will receive reduce conflicts, but that is okay as it won't hinder the working of the code.

gcc lex.yy.c -ll

3.Run the program using the command:

./a.out

Output :

```
 C code:
int a    ; int b    ; int c    ;     a = 10    ;     b = 15    ; While (
t1 = a + b    ;     t1>20    ; )  {If(         b>15    ; ) {         t2 = a + b
   ; c = t2    ;  }  } printf(" %d   " ,c);   myfunction(int a   );   int myfunc
tion(int a   ){ printf(" %d   " ,c);   }
```