Compiler Design Lab-5

Intermediate Code Generation

Name: Arjun N R

SRN: PES2UG22CS910

Lexer.l

```
%{
   #define YYSTYPE char*
   #include <unistd.h>
   #include "y.tab.h"
   #include <stdio.h>
   extern void yyerror(const char *); // declare the error handling function
%}

/* Regular definitions */
digit   [0-9]
letter  [a-zA-Z]
id      {letter}({letter}|{digit})*
digits  {digit}+
opFraction   (\.{digits})?
opExponent ([Ee][+-]?{digits})?
number      {digits}{opFraction}{opExponent}
%option yylineno

%%
\/\/(.*) ; // ignore comments
```

```
[\t\n] ; // ignore whitespaces
"("             {return *yytext;}
")"             {return *yytext;}
"."             {return *yytext;}
","             {return *yytext;}
"*"             {return *yytext;}
"+"             {return *yytext;}
";"             {return *yytext;}
"-"             {return *yytext;}
"/"             {return *yytext;}
"="             {return *yytext;}
">"             {return *yytext;}
"<"             {return *yytext;}
{number}    {
                    yylval = strdup(yytext);  //stores the value of the number to be used later for symbol table insertion
                    return T_NUM;
                }
{id}            {
                                    yylval = strdup(yytext); //stores the identifier to be used later for symbol table insertion
                                    return T_ID;
                                }
.               {} // anything else => ignore
%%

int yywrap() {
```

```
    return 1;
}
```

Parser.y

```
%{
    #include "quad_generation.h"
    #include <stdio.h>
    #include <stdlib.h>
    #include <string.h>

    #define YYSTYPE char*

    void yyerror(char* s);
    int yylex();
    extern int yylineno;
    extern FILE* yyin;

    FILE* icg_quad_file;
    int temp_no = 1;
%}

%token T_ID T_NUM

%start START

%%
```

```
START : ASSGN   {
                printf("Valid syntax\n");
                YYACCEPT;
              }

ASSGN : T_ID '=' E  { quad_code_gen($1, $3, "=", NULL); }

E : E '+' T     { char* temp = new_temp(); quad_code_gen(temp, $1, "+", $3); $$ = temp; }
  | E '-' T     { char* temp = new_temp(); quad_code_gen(temp, $1, "-", $3); $$ = temp; }
  | T           { $$ = $1; }

T : T '*' F     { char* temp = new_temp(); quad_code_gen(temp, $1, "*", $3); $$ = temp; }
  | T '/' F     { char* temp = new_temp(); quad_code_gen(temp, $1, "/", $3); $$ = temp; }
  | F           { $$ = $1; }

F : '(' E ')'   { $$ = $2; }
  | T_ID        { $$ = $1; }
  | T_NUM       { $$ = $1; }
%%

void yyerror(char* s)
{
   printf("Error :%s at %d \n",s,yylineno);
```

```
}

int main(int argc, char* argv[])
{
    if (argc < 2) {
        printf("Usage: %s <input_file>\n", argv[0]);
        return 1;
    }

    FILE* input_file = fopen(argv[1], "r");
    if (!input_file) {
        perror("Error opening input file");
        return 1;
    }

    yyin = input_file; // Set the input stream for the lexer
    icg_quad_file = fopen("icg_quad.txt", "w");
    if (!icg_quad_file) {
        perror("Error opening output file");
        fclose(input_file);
        return 1;
    }

    yyparse();

    fclose(input_file);
```

```
    fclose(icg_quad_file);
    return 0;
}
```

Quad_generation.h

```
#ifndef QUAD_GENERATION_H
#define QUAD_GENERATION_H

#include <stdio.h>

extern FILE* icg_quad_file;  // Pointer to the output file for quadruples
extern int temp_no;          // Variable to keep track of current temporary count

void quad_code_gen(char* a, char* b, char* op, char* c);
char* new_temp();

#endif
```

Quad_generation.c

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "quad_generation.h"

void quad_code_gen(char* a, char* b, char* op, char* c)
{
    if (c == NULL || strlen(c) == 0) {
```

```
        fprintf(icg_quad_file, "%s, %s, , %s\n", op, b, a);
    } else {
        fprintf(icg_quad_file, "%s, %s, %s, %s\n", op, b, c, a);
    }
}


char* new_temp()
{
    char* temp = (char*)malloc(sizeof(char)*4);
    sprintf(temp, "t%d", temp_no);
    ++temp_no;
    return temp;
}
```

Output

```
PS C:\Users\arjun\Documents\SEM-6\CD\CompilerDesign\Lab5\Lab 5 (ICG)> ./a.exe .\test_input_2.c
Valid syntax
PS C:\Users\arjun\Documents\SEM-6\CD\CompilerDesign\Lab5\Lab 5 (ICG)> ./a.exe .\test_input_1.c
Valid syntax
```

icg_quad2.txt

Lab 5 (ICG) > icg_quad2.txt

```
1   /, c, 6.7, t1
2   +, t1, 12.45, t2
3   *, a, 1234.0, t3
4   -, t2, t3, t4
5   =, t4, , b
```

icg_quad.txt

Lab 5 (ICG) > icg_quad.txt

```
1   /, 9, 2, t1
2   +, t1, a, t2
3   -, t2, b, t3
4   =, t3, , x
```