BCSE307P
Compiler Design Lab
Lab Assignment 4

Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)
CHENNAI

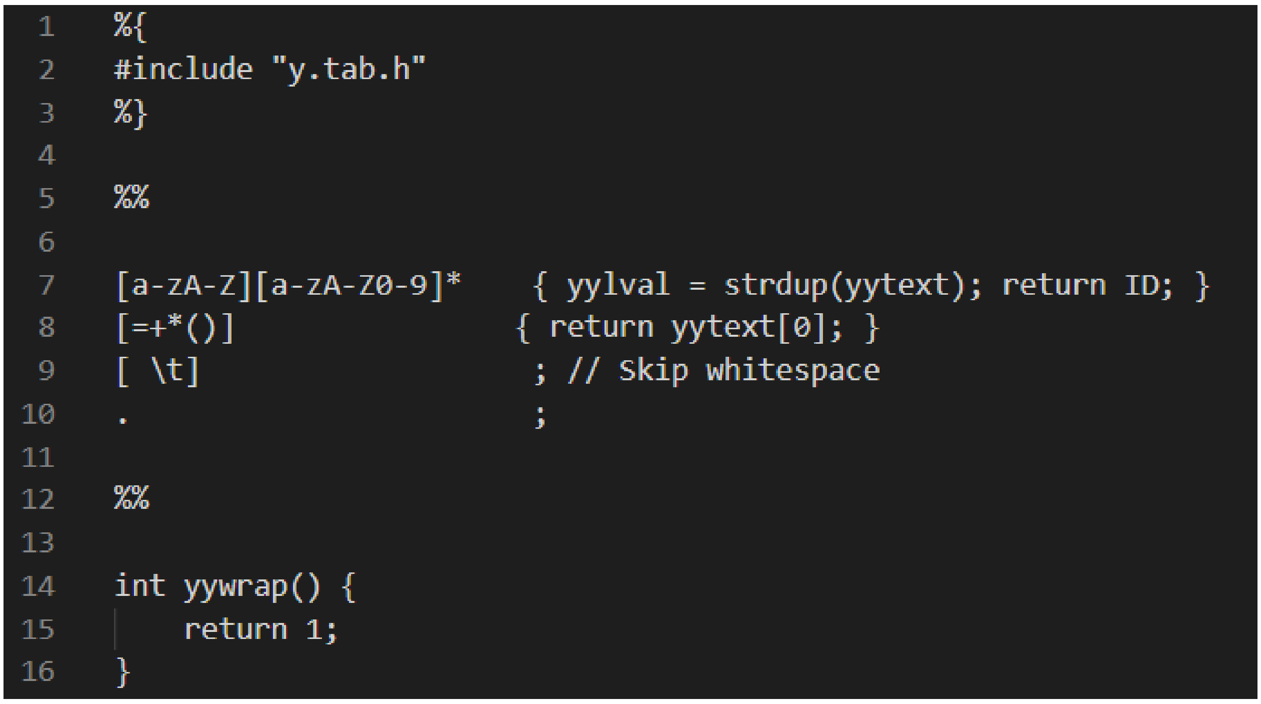
Name – Ishan Kapoor

Registration Number – 21BCE5882

Submitted to – Prof. S. Srisakthi

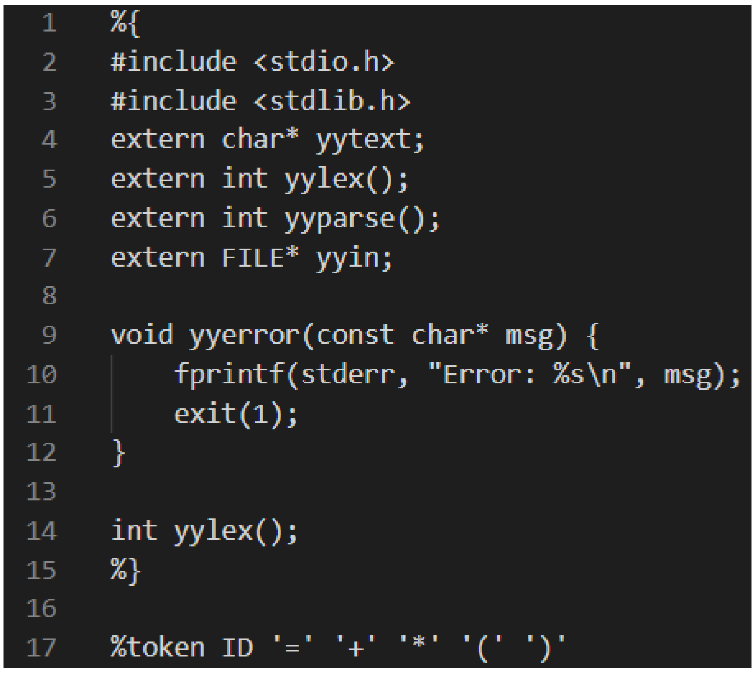
Program to convert the two expressions into a three-address code.

- Lex Code:

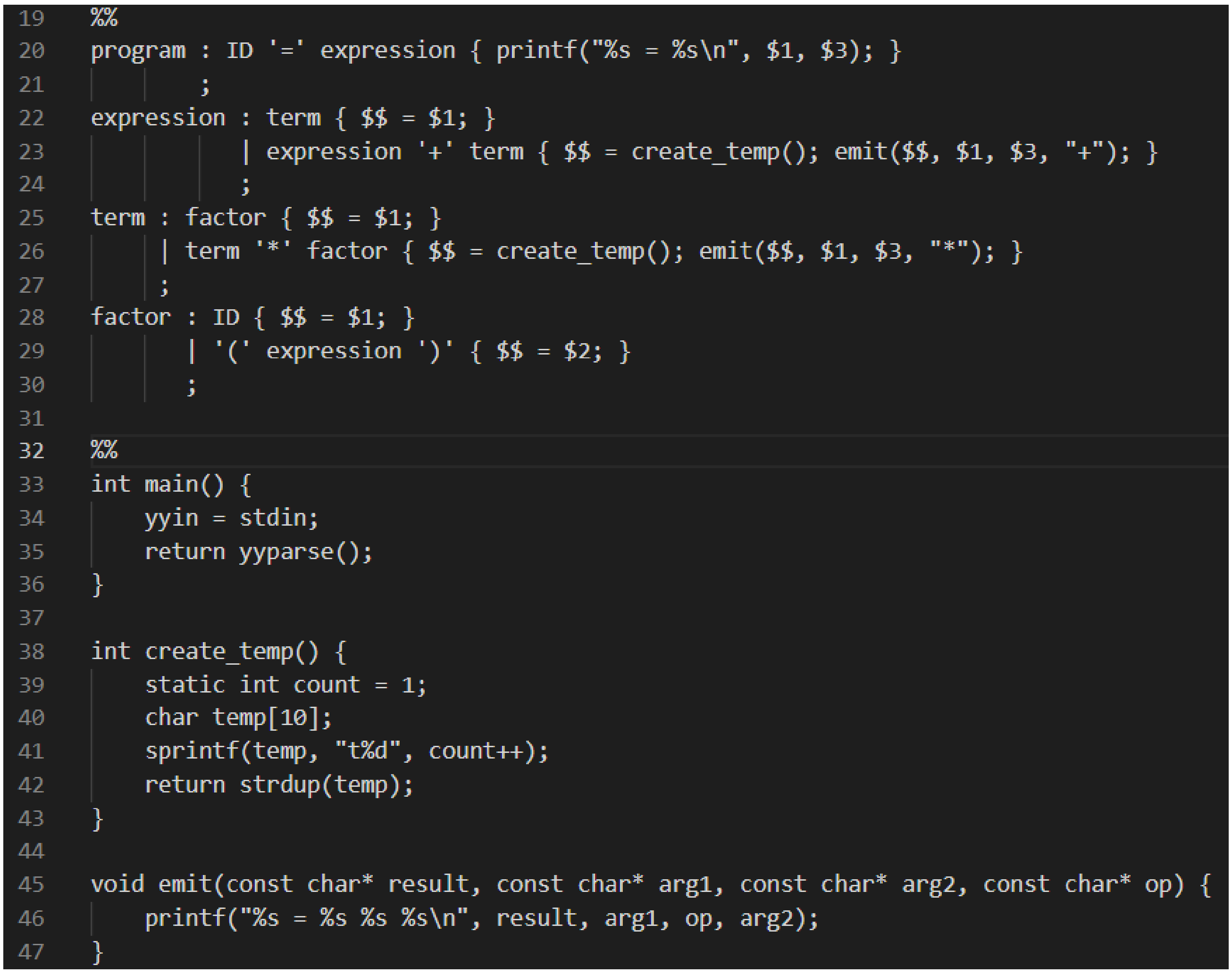


```
%{
#include "y.tab.h"
%}

%%

[a-zA-Z][a-zA-Z0-9]*    { yylval = strdup(yytext); return ID; }
[=+*()]                { return yytext[0]; }
[ \t]                   ; // Skip whitespace
.                       ;

%%

int yywrap() {
    return 1;
}
```

- Yacc Code:

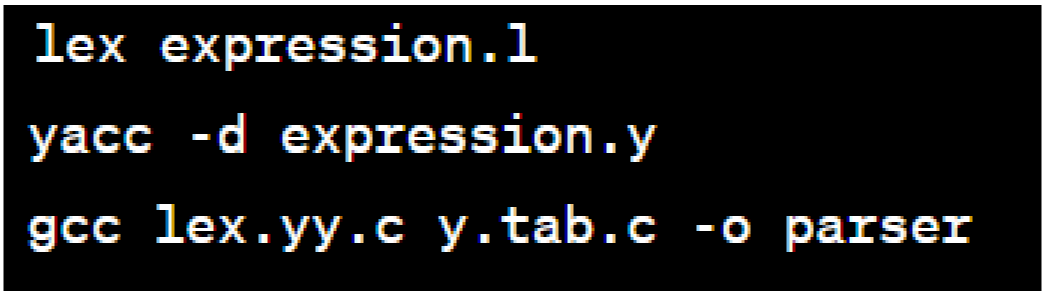


```
%{
#include <stdio.h>
#include <stdlib.h>
extern char* yytext;
extern int yylex();
extern int yyparse();
extern FILE* yyin;

void yyerror(const char* msg) {
    fprintf(stderr, "Error: %s\n", msg);
    exit(1);
}

int yylex();
%}

%token ID '=' '+' '*' '(' ')'
```



```
%%
program : ID '=' expression { printf("%s = %s\n", $1, $3); }
        ;
expression : term { $$ = $1; }
           | expression '+' term { $$ = create_temp(); emit($$, $1, $3, "+"); }
           ;
term : factor { $$ = $1; }
     | term '*' factor { $$ = create_temp(); emit($$, $1, $3, "*"); }
     ;
factor : ID { $$ = $1; }
       | '(' expression ')' { $$ = $2; }
       ;

%%
int main() {
    yyin = stdin;
    return yyparse();
}

int create_temp() {
    static int count = 1;
    char temp[10];
    sprintf(temp, "t%d", count++);
    return strdup(temp);
}

void emit(const char* result, const char* arg1, const char* arg2, const char* op) {
    printf("%s = %s %s %s\n", result, arg1, op, arg2);
}
```

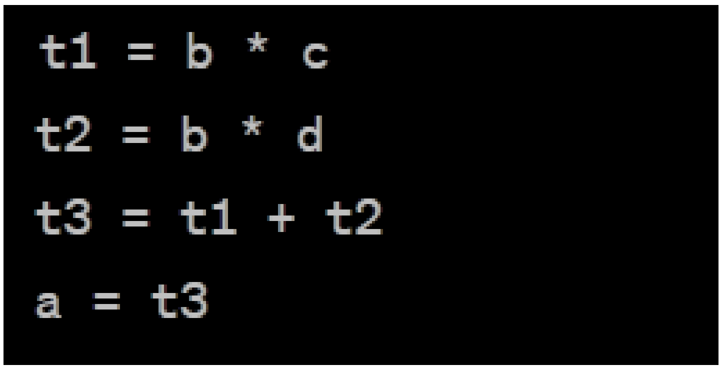
- Compiling Process



```
lex expression.l

yacc -d expression.y

gcc lex.yy.c y.tab.c -o parser
```

Expression: a = b*c + b*d

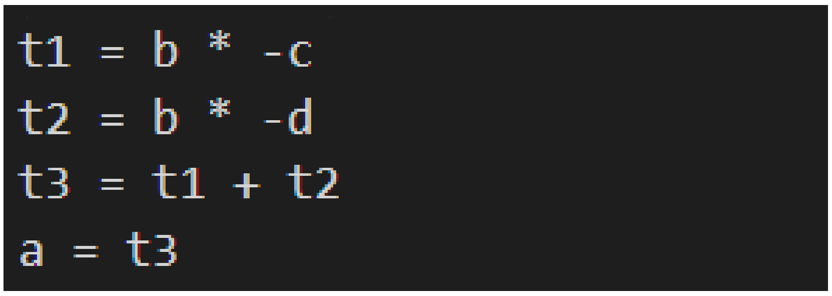
Output:



```
t1 = b * c
t2 = b * d
t3 = t1 + t2
a = t3
```

Expression: a = b*-c + b*-d

Output:



```
t1 = b * -c
t2 = b * -d
t3 = t1 + t2
a = t3
```