**TASK12:write a program to implement 3 address code.**

%{

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int tempVarCount = 0; // Counter for temporary variables

char tempVar[10]; // Temporary variable name buffer

// Function to generate the next temporary variable name

void generateTempVar() {

sprintf(tempVar, "t%d", tempVarCount++);

}

%}

DIGIT [0-9]

ID [a-zA-Z][a-zA-Z0-9]\*

%%

{ID} { printf("PUSH %s\n", yytext); } // Push identifier onto the stack

{DIGIT}+ { printf("PUSH %s\n", yytext); } // Push number onto the stack

"+" { printf("POP B\nPOP A\nADD A, B\n"); // Handle addition

generateTempVar();

printf("PUSH %s\n", tempVar); }

"-" { printf("POP B\nPOP A\nSUB A, B\n"); // Handle subtraction

generateTempVar();

printf("PUSH %s\n", tempVar); }

"\*" { printf("POP B\nPOP A\nMUL A, B\n"); // Handle multiplication

generateTempVar();

printf("PUSH %s\n", tempVar); }

"/" { printf("POP B\nPOP A\nDIV A, B\n"); // Handle division

generateTempVar();

printf("PUSH %s\n", tempVar); }

"(" { /\* Ignore opening parentheses \*/ }

")" { /\* Ignore closing parentheses \*/ }

[ \t\n]+ { /\* Ignore whitespace \*/ }

. { printf("Invalid character: %s\n", yytext); exit(1); } // Handle unexpected characters

%%

int main(int argc, char \*\*argv) {

if (argc > 1) {

FILE \*file = fopen(argv[1], "r");

if (!file) {

perror(argv[1]);

return 1;

}

yyin = file; // Redirect input to file

}

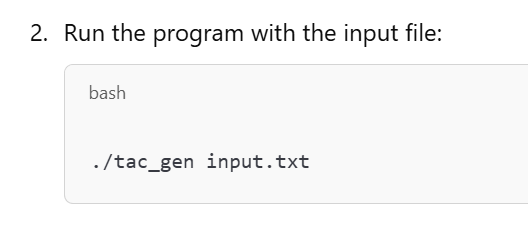
printf("Three-Address Code Generation:\n");

yylex();

return 0;

}





Output:

Three-Address Code Generation:

PUSH a

PUSH b

PUSH c

POP B

POP A

MUL A, B

PUSH t0

POP B

POP A

ADD A, B

PUSH t1