**PROGRAMMING ASSIGNMENT 6 –** Generate three address code for a simple program using LEX and YACC

**tac.l**

%{

#include<stdio.h>

#include<string.h>

#include "y.tab.h"

%}

term ([a-zA-Z\\_][a-zA-Z\\_0-9]\*|[0-9]+)

relop ("<"|"<="|">"|">="|"=="|"!=")

op ("+"|"-"|"\*"|"/"|"%")

%%

"while" { return WHILE; }

"do" { return DO; }

"switch" { return SWITCH; }

"case" { return CASE; }

"default" { return DEFAULT; }

"break" { return BREAK; }

{term} { yylval.str = strdup(yytext); return TERM; }

{relop} { yylval.str = strdup(yytext); return RELOP; }

{op} { yylval.str = strdup(yytext); return OP; }

[ \t\n]+ { }

. { return \*yytext; }

**tac.y**

%{

#include<stdio.h>

#include<stdlib.h>

#include<math.h>

int yylex(void);

#include "y.tab.h"

extern FILE \*yyin;

int cc = 1, tc = 1, sc = 0, currcase = 1;

%}

%token TERM RELOP OP WHILE DO SWITCH CASE DEFAULT BREAK

%union

{

int intval;

float floatval;

char \*str;

}

%type<str> TERM RELOP OP

%%

line: /\* empty \*/

| TERM '=' TERM OP TERM ';' { printf("t%d := %s %s %s\n%s := t%d\n", tc, $3, $4, $5, $1, tc); tc++; } line

| TERM '=' TERM RELOP TERM ';' { printf("t%d := %s %s %s\n%s := t%d\n", tc, $3, $4, $5, $1, tc); tc++; } line

| TERM '=' TERM ';' { printf("%s := %s\n", $1, $3); } line

| WHILE TERM RELOP TERM DO '{' { printf("LABEL%d: if not %s %s %s goto FALSE%d\nTRUE%d: ", cc, $2, $3, $4, cc, cc); } line '}' { printf("FALSE%d: ", cc); cc++; } line

| WHILE TERM OP TERM DO '{' { printf("LABEL%d: if not %s %s %s goto FALSE%d\nTRUE%d: ", cc, $2, $3, $4, cc, cc); } line '}' { printf("FALSE%d: ", cc); cc++; } line

| WHILE TERM DO '{' { printf("LABEL%d: if not %s then goto FALSE%d\nTRUE%d: ", cc, $2, cc, cc); } line '}' { printf("FALSE%d:", cc); cc++; } line

| SWITCH '(' TERM RELOP TERM ')' '{' { printf("t%d := %s %s %s\n", tc, $3, $4, $5); sc = tc; tc++; } cases '}' { printf("NEXT%d: ", cc); cc++; } line

| SWITCH '(' TERM OP TERM ')' '{' { printf("t%d := %s %s %s\n", tc, $3, $4, $5); sc = tc; tc++; } cases '}' { printf("NEXT%d: ", cc); cc++; } line

| SWITCH '(' TERM ')' '{' { printf("t%d := %s\n", tc, $3); sc = tc; tc++; } cases '}' { printf("NEXT%d: ", cc); cc++; } line

| BREAK ';' line { printf("goto NEXT%d\n", cc); }

cases: /\* empty \*/

| CASE TERM ':' { printf("CASE%d: if t%d != %s goto CASE%d\n ",currcase, sc, $2,currcase+1); currcase++; } line cases

| DEFAULT{printf("CASE%d: ",currcase);} ':' line { printf("goto NEXT%d\n", cc); } cases

%%

int yyerror(char\* s)

{

fprintf(stderr, "%s\n", s);

return 0;

}

int yywrap()

{

return 1;

}

int main()

{

char inputFile[100];

printf("Enter the input file: ");

scanf("%s",inputFile);

yyin = fopen(inputFile, "r");

yyparse();

printf("\n");

return 0;

}

**input.in**

while i < 10 do

{

a =0;

i = i +1;

}

switch(i+j) {

case 1: x = y + z; break;

case 2: u = v + w; break;

default: p = q + r;

}

a = 5;

**Output:**

