VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

B.Tech. Winter Semester 2023-24

School Of Computer Science and Engineering (SCOPE)

Digital Assignment - V

Compiler Design Lab

Apurva Mishra, 22BCE2791

21 October 2024

Note: This is mirror of Assessment 5 submitted on 21 October, 2024 on Moodle

1. Questions

Problem 1.1.
Write a C program to generate 3-Address code for a given expression

Code

main.l

```
%{
#include <stdlib.h>
#include <stdio.h>
void yyerror(char *);
#include "y.tab.h"
%}
%%
[0-9]+ {
yylval = atoi(yytext);
return INTEGER;
}
[-+/*\n] { return *yytext; }
[ \t] ;
. yyerror("invalid character");
%%
int yywrap(void) {
return 1;
}
```

main.y

```
token INTEGER VARIABLE
%left '+' '-'
%left '*' '/'
%{
#include <stdio.h>
void yyerror(char *);
int yylex(void);
int sym[26];
int count = 0;
%}
%%
program:
program statement '\n'
|
;
statement:
expr { ;}
|
;
expr:
INTEGER
```

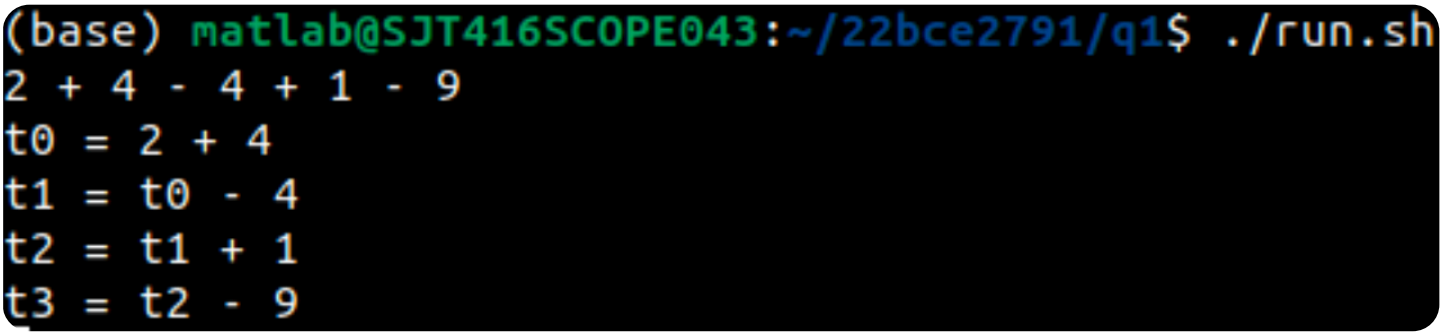
```
| expr '+' expr { $$ = $1 + $3; if (count == 0) {printf("t%d = %d + %d\n", count, $1, $3);} else
{printf("t%d = t%d + %d\n", count, count-1, $3);} count++;}
| expr '-' expr { $$ = $1 - $3; if (count == 0) {printf("t%d = %d - %d\n", count, $1, $3);} else
{printf("t%d = t%d - %d\n", count, count-1, $3);} count++;}
;
%%
void yyerror(char *s) {
fprintf(stderr, "%s\n", s);
}
int main(void) {
yyparse();
return 0;
}
```

run.sh

```
#!/bin/bash

lex main.l
yacc -d main.y
gcc lex.yy.c y.tab.c -o main
./main
```

Output



Problem 1.2.
Write a C Program to implement Type Checking

Code

main.l

```
%{
#include <stdlib.h>
#include <stdio.h>
void yyerror(char *);
#include "y.tab.h"
%}
%%
("int") {return TINTEGER; }
("char") {return TCHAR; }
[0-9]+ {
yylval = atoi(yytext);
return INTEGER;
}
[a-z]+ { return WORD; }
[=] { return *yytext; }
[ \t] ;
. yyerror("invalid character");
%%
int yywrap(void) {
return 1;
}
```

main.y

```
%token TINTEGER TCHAR INTEGER WORD
%left '='
%{
#include <stdio.h>
void yyerror(char *);
int yylex(void);
int sym[26];
%}
%%
program:
program statement '\n'
|
;
statement:
expr { ;}
|
;
expr:
INTEGER
| TINTEGER WORD'=' INTEGER {$$ = 1;printf("Valid type");}
| TCHAR WORD '=' WORD {$$ = 1;printf("Valid type");}
;
```

```
%%
void yyerror(char *s) {
fprintf(stderr, "%s\n", s);
}
int main(void) {
yyparse();
return 0;
}
```

run.sh

```
#!/bin/bash
lex main.l
yacc -d main.y
gcc lex.yy.c y.tab.c -o main
./main
```

Output