19BCE248
2CS701
Compiler Construction

## Practical 6

**Aim :-** Intermediate Code Generation - To generate Three Address code for assignment statement.

**Code :-**

**Prac6.l**

```
%{
#include <stdio.h>
#include <stdlib.h>
#include "y.tab.h"
%}
%%
[0-9]+ {yylval.symbol = yytext[0]; return NUMBER;}
[a-zA-z]+ {yylval.symbol=yytext[0]; return LETTER;}
\n {return 0;}
. {return yytext[0];}
%%
yywrap(){
return 1;
}
```

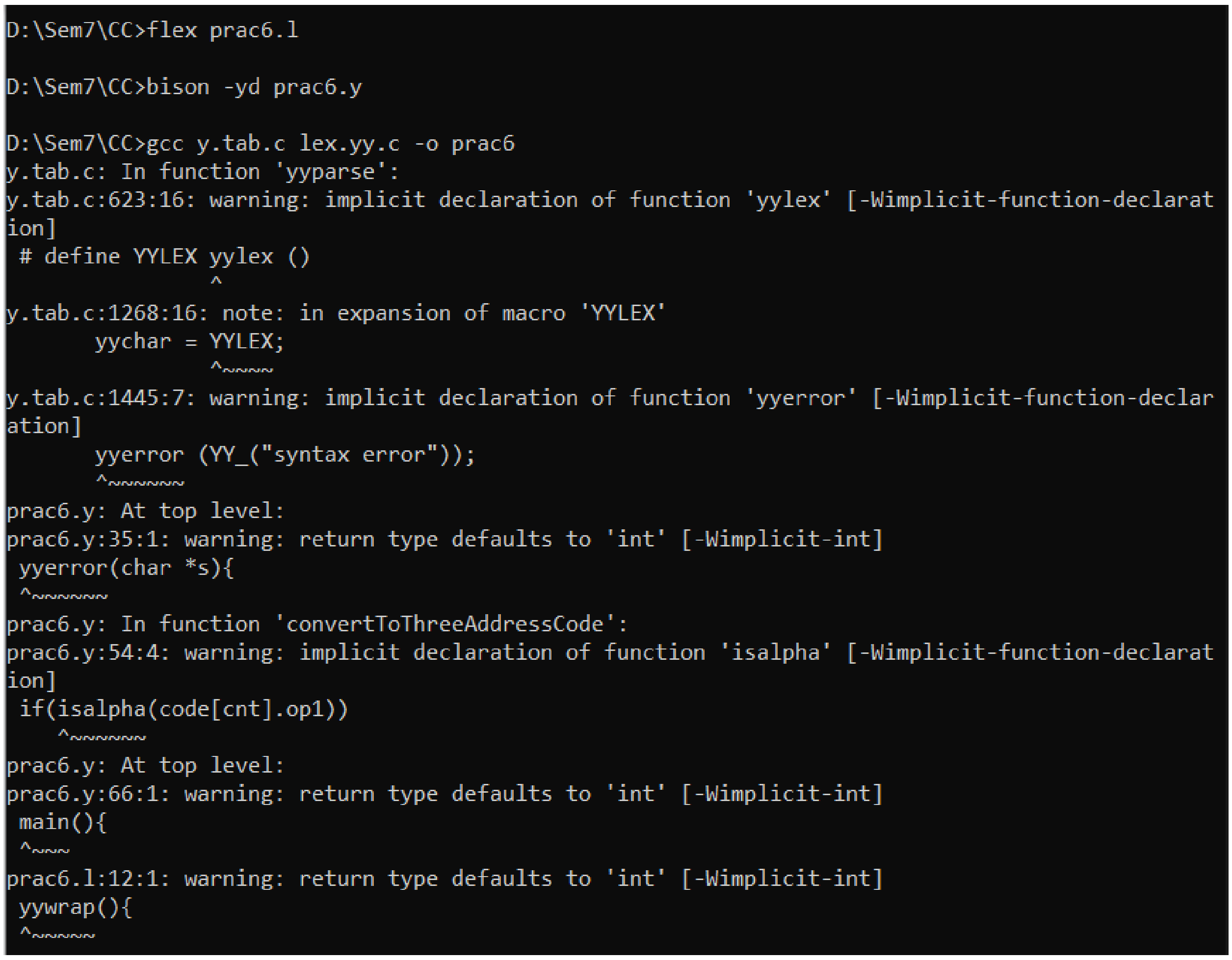
**Prac6.y**

```
%{
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
void convertToThreeAddressCode();
char addToTable(char,char,char);
int i = 0;
char tmp='1';
struct exp{
char op1,op2,op;
};
%}
%union
```

```
{
char symbol;
}
%token <symbol> LETTER NUMBER
%type <symbol> e
%left '+' '-'
%left '*' '/' '%'
%%
stmt: LETTER '=' e ';' {addToTable($1,'=',$3);}
|e ';'
;
e: e '/' e {$$ = addToTable($1,'/',$3);}
| e '*' e {$$ = addToTable($1,'*',$3);}
| e '%' e {$$ = addToTable($1,'%',$3);}
| e '+' e {$$ = addToTable($1,'+',$3);}
| e '-' e {$$ = addToTable($1,'-',$3);}
| '(' e ')' {$$ = (char)$2;}
| NUMBER {$$=$1;}
| LETTER {$$=$1;}
;
%%
yyerror(char *s){
printf("%s",s);
exit(0);
}
struct exp code[20];
char addToTable(char op1,char op,char op2){
code[i].op1=op1;
code[i].op=op;
code[i].op2=op2;
i++;
return tmp++;
}
void convertToThreeAddressCode(){
printf("\nThree Address Code\n\n");
int cnt=0;
char tmp='1';
while(cnt < i){
if(code[cnt].op != '=')
printf("t%c = ",tmp++);
if(isalpha(code[cnt].op1))
printf("%c ",code[cnt].op1);
else if(code[cnt].op1 >='1' && code[cnt].op1 <='9')
printf("t%c ",code[cnt].op1);
printf("%c ",code[cnt].op);
if(isalpha(code[cnt].op2))
printf("%c \n",code[cnt].op2);
else if(code[cnt].op2 >='1' && code[cnt].op2 <='9')
printf("t%c \n",code[cnt].op2);
cnt++;
}
}
```

```
main(){
printf("\nEnter the expression: ");
yyparse();
convertToThreeAddressCode();
}
```

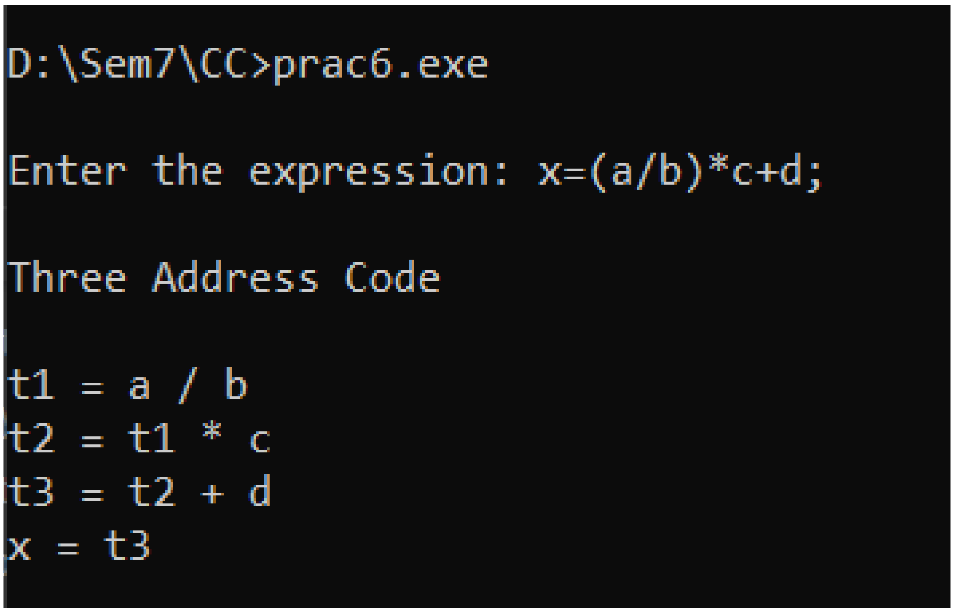
## Output:



```
D:\Sem7\CC>flex prac6.l

D:\Sem7\CC>bison -yd prac6.y

D:\Sem7\CC>gcc y.tab.c lex.yy.c -o prac6
y.tab.c: In function 'yyparse':
y.tab.c:623:16: warning: implicit declaration of function 'yylex' [-Wimplicit-function-declarat
ion]
 # define YYLEX yylex ()
                ^
y.tab.c:1268:16: note: in expansion of macro 'YYLEX'
       yychar = YYLEX;
                ^~~~~
y.tab.c:1445:7: warning: implicit declaration of function 'yyerror' [-Wimplicit-function-declar
ation]
       yyerror (YY_("syntax error"));
       ^~~~~~~
prac6.y: At top level:
prac6.y:35:1: warning: return type defaults to 'int' [-Wimplicit-int]
 yyerror(char *s){
 ^~~~~~~
prac6.y: In function 'convertToThreeAddressCode':
prac6.y:54:4: warning: implicit declaration of function 'isalpha' [-Wimplicit-function-declarat
ion]
 if(isalpha(code[cnt].op1))
    ^~~~~~~
prac6.y: At top level:
prac6.y:66:1: warning: return type defaults to 'int' [-Wimplicit-int]
 main(){
 ^~~~
prac6.l:12:1: warning: return type defaults to 'int' [-Wimplicit-int]
 yywrap(){
 ^~~~~~
```



```
D:\Sem7\CC>prac6.exe

Enter the expression: x=(a/b)*c+d;

Three Address Code

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