

☐ Remember me?[Register](#)[Forgot Your Password?](#)

- [HOME](#)
- [About Us](#)
- [Contact Us](#)
- [Contribute Your Stuff](#)



- [COMPUTER ENGG »](#)
- [INFORMATION TECHNOLOGY. »](#)
- [BCA »](#)
- [VIVA »](#)
- [PROJECT ASSISTANCE »](#)
- [OTHER CODES »](#)
- [Free Online Tests »](#)

2013/10/26 [BE \(COMP\)](#), [lex-yacc](#)

Generate three address code ,quadruple and triple using lex and yacc

/* Code to generate three address code ,quadruple and triple using lex and yacc */

[download source code and output](#)

CODE :

/* LEX FILE */

```
%{
#include "y.tab.h"
extern char yyval;
%}
```

```
NUMBER [0-9]+
LETTER [a-zA-Z]+
```

```
%%
```

```
{NUMBER} {yyval.sym=(char)yytext[0]; return NUMBER;}
{LETTER} {yyval.sym=(char)yytext[0];return LETTER;}
```

```
\n {return 0;}
. {return yytext[0];}
```

```
%%
```

```
/* yacc file */
```

```
%{
```

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
void ThreeAddressCode();
void triple();
void qudruple();
char AddToTable(char ,char, char);
```

```
int ind=0;
char temp='A';
struct incod
{
char opd1;
char opd2;
char opr;
};
%}
```

```
%union
{
char sym;
}
```

```
%token <sym> LETTER NUMBER
%type <sym> expr
%left '-' '+'
%right '*' '/'
```

```
%%
```

```
statement: LETTER '=' expr ';' {AddToTable((char)$1,(char)$3,'=');}
| expr ';'
;
```

```
expr: expr '+' expr {$$ = AddToTable((char)$1,(char)$3,'+');}
| expr '-' expr {$$ = AddToTable((char)$1,(char)$3,'-');}
| expr '*' expr {$$ = AddToTable((char)$1,(char)$3,'*');}
| expr '/' expr {$$ = AddToTable((char)$1,(char)$3,'/');}
| '(' expr ')' {$$ = (char)$2;}
| NUMBER {$$ = (char)$1;}
| LETTER {$$ = (char)$1;}
;
```

```
%%
```

```
yyerror(char *s)
{
printf("%s",s);
exit(0);
}
```

```
struct incod code[20];
```

```
int id=0;
```

```
char AddToTable(char opd1,char opd2,char opr)
{
code[ind].opd1=opd1;
code[ind].opd2=opd2;
code[ind].opr=opr;
ind++;
temp++;
return temp;
}
```

```
void ThreeAddressCode()
{
int cnt=0;
temp++;
printf("\n\n\t THREE ADDRESS CODE\n\n");
while(cnt<ind)
{
printf("%c : = \t",temp);
if(isalpha(code[cnt].opd1))
printf("%c\t",code[cnt].opd1);
else
{printf("%c\t",temp);}

printf("%c\t",code[cnt].opr);

if(isalpha(code[cnt].opd2))
printf("%c\t",code[cnt].opd2);
else
{printf("%c\t",temp);}

printf("\n");
cnt++;
temp++;
}
}
```

```
void quadruple()
{
int cnt=0;
temp++;
printf("\n\n\t QUADRUPLE CODE\n\n");
while(cnt<ind)
{
//printf("%c : = \t",temp);
printf("%d",id);
printf("\t");
printf("%c",code[cnt].opr);
printf("\t");
if(isalpha(code[cnt].opd1))
printf("%c\t",code[cnt].opd1);
else
{printf("%c\t",temp);}

//printf("%c\t",code[cnt].opr);
```

```
if(isalpha(code[cnt].opd2))
printf("%c\t",code[cnt].opd2);
else
{printf("%c\t",temp);}

printf("%c",temp);

printf("\n");
cnt++;
temp++;
id++;

}
}

void triple()
{
int cnt=0,cnt1,id1=0;
temp++;
printf("\n\n\t TRIPLE CODE\n\n");
while(cnt<ind)
{
//printf("%c : = \t",temp);

if(id1==0)
{
printf("%d",id1);
printf("\t");
printf("%c",code[cnt].opr);
printf("\t");
if(isalpha(code[cnt].opd1))
printf("%c\t",code[cnt].opd1);
else
{printf("%c\t",temp);}

//printf("%c\t",code[cnt].opr);
cnt1=cnt-1;
if(isalpha(code[cnt].opd2))
printf("%c",code[cnt].opd2);
else
{printf("%c\t",temp);}
}
else
{
printf("%d",id1);
printf("\t");
printf("%c",code[cnt].opr);
printf("\t");
if(isalpha(code[cnt].opd1))
printf("%c\t",code[cnt].opd1);
else
{printf("%c\t",temp);}

//printf("%c\t",code[cnt].opr);
cnt1=cnt-1;
if(isalpha(code[cnt].opd2))
```

```

printf("%d",id1-1);
else
{printf("%c\t",temp);}
}

printf("\n");
cnt++;
temp++;
id1++;

}

}

main()
{
printf("\nEnter the Expression: ");
yyparse();
temp='A';
ThreeAddressCode();
quadruple();
triple();
}

yywrap()
{
return 1;
}

```

OUTPUT

```

administrator@ubuntu:~/Desktop$ flex th.l
administrator@ubuntu:~/Desktop$ yacc -d th.y
administrator@ubuntu:~/Desktop$ gcc lex.yy.c y.tab.c -ll -lm
administrator@ubuntu:~/Desktop$ ./a.out

```

Enter the Expression: a=((b+c)*(d+e))
syntax error

```
administrator@ubuntu:~/Desktop$ ./a.out
```

Enter the Expression: a=((b+c)*(d/e));
THREE ADDRESS CODE

```

B := b + c
C := d / e
D := B * C
E := a = D
QUADRUPLE CODE

```

```

0 + b c G
1 / d e H
2 * B C I
3 = a D J
TRIPLE CODE

```

0 + b c

1 / d 0

2 * B 1

3 = a 2

administrator@ubuntu:~/Desktop\$

• SEARCH – CodeArea

•

• Like Us

• Recent Posts

- [C code to find minimum element of an array using recursion function \(Divide And Conquer\)](#)
- [JAVA](#)
- [Java code to find and replace text](#)
- [Write a Java program to accept list of files as command line arguments. Display the name and size of all the files. Delete all files with extension as '.html' from the current directory. Appropriate error messages should be printed.\(Use vector Array\)](#)
- [Java code for User defined exceptions](#)

• Ad

• Recent Comments

- pswebsolutions on [Networking VIVA Questions](#)
- Vishal Motwani on [VB PROJECT GUIDANCE](#)
- pswebsolutions on [Project in VB \(visual basic \)](#)
- kalyani supekar on [Project in VB \(visual basic \)](#)
- [Mandi](#) on [Circular-queue linked list](#)

• Archives

- [August 2015](#)
- [September 2014](#)
- [August 2014](#)
- [July 2014](#)
- [January 2014](#)
- [October 2013](#)
- [July 2013](#)
- [March 2013](#)
- [January 2013](#)
- [December 2012](#)
- [October 2012](#)
- [September 2012](#)
- [August 2012](#)
- [July 2012](#)

• Live Traffic Feed

[Live Traffic Feed](#)

A visitor from India viewed "[lex-yacc | Code Area](#)" 40 mins ago
 A visitor from New Delhi, Delhi viewed "[Memory allocation algorithms \(First Fit, Best Fit , Next Fit \) | Code Area](#)" 41 mins ago
 A visitor from Shadnagar, Andhra Pradesh viewed "[Networking VIVA Questions | Code Area](#)" 45 mins ago
 A visitor from Port Louis viewed "[CLASS DIAGRAM for Library Management System | Code Area](#)" 54 mins ago
 A visitor from West Bengal viewed "[Producer-Consumer problem\(using semaphore \) | Code Area](#)" 57 mins ago
 A visitor from Pune, Maharashtra viewed "[TIME TABLE FOR SECOND YEAR ENGINEERING \(Pune University\) | Code Area](#)" 1 hr 1 mins ago
 A visitor from India viewed "[String Operations using Lex | Code Area](#)" 1 hr 3 mins ago
 A visitor from India viewed "[CLASS DIAGRAM for Library Management System | Code Area](#)" 1 hr 4 mins ago
 A visitor from Mountain View, California viewed "[Data Structures | Code Area](#)" 1 hr 6 mins ago
 A visitor from Jaipur, Rajasthan viewed "[Code for Telephone Directory \(Java\) | Code Area](#)" 1 hr 7

[Real-time view](#) • [Get Feedjit](#)

[Feedjit Widget](#)

Copyright © 2012 CodeArea | Powered by CodeArea
 Powered by [CodeArea](#) | Designed by: [CodeArea](#)