Nikit Gokhe

Class : TY-Comp D1

Roll No. 324022

GR No. 21810522

# Assignment 05

# Aim:

Write a program for converting a simple expression into 3 address code.

# Theory:

Three address code is a type of intermediate code which is easy to generate and can be easily converted to machine code.It makes use of at most three addresses and one operator to represent an expression and the value computed at each instruction is stored in temporary variable generated by compiler. The compiler decides the order of operation given by three address code. GENERAL REPRESENTATION : a = b op c a,b,c are identifiers op is the operator.

# Source Code:

## ass5.l

%{

#include

“y.tab.h” extern

Int yylval;

%}

%%

[A-Za-z]([A-Za-z][0-9])\* return ID; [0-9]+ {yylval=atoi(yytext);return NUM;} [\n\t] yyterminate();

return yytext[0];

%%

## ass5.y

%{

#include void yyerror(char\*); int yylex(void);

char n

='A'; int i=0;

%}

%token NUM

%token ID

%left '+''-''\*''/'

%nonassoc UMINUS

%%

S : E {printf("x=%c\n",n-

1);} ; E: NUM{} |

E'+'E{if(i==0){ printf("%c = %d %c %d\n",n,$1,'+',$3); n++;i+ +;}

else{ printf("%c = %c %c %d\n",n,n-1,'+',$3);n++;}}

|E'-'E{if(i==0){ printf("%c = %d %c %d\n",n,$1,'-',$3); n++;i+ +;}

else{ printf("%c = %c %c %d\n",n,n-1,'-',$3);n++;}} |

E'\*'E{if(i==0){ printf("%c = %d %c %d\n",n,$1,'\*',$3); n++;i+ +;}

else{ printf("%c = %c %c %d\n",n,n-1,'\*',$3);n++;}}

|E'/'E{if(i==0){ printf("%c = %d %c %d\n",n,$1,'/',$3); n++;i+ +;}

else{ printf("%c = %c %c %d\n",n,n-1,'/',$3);n++;}} ;

%%

void yyerror(char\* s)

{

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

}

int yywrap()

{

return 1;

}

int main()

{

printf("Enter Expression x => "); yyparse();

yywrap();

return 0;

}

# Output:

nikit@user:~$ ass5.l

nikit@user:~$ yacc-d ass5.y

nikit@user:~$ gcc lex.yy.c y.tab.c

nikit@user:~$ ./a.out

Enter Expression x => 9/3\*2+5-4

A = 9 / 3

B = A \* 2

C = B + 5

D = C – 4

x =D