**Name :- Aboli Satish Jadhav**

**Class :- TY-C3**

**Roll No :- 323082**

**GR No. :- 22020258**

==========

# Assignment 4

**Task 1 :-** Evaluate an Arithmetic Expression using YACC tool.

## Code :-

- **4A_A4.l**

```
%{
#include<stdio.h>
#include "y.tab.h"
%}
%%
[0-9]+          {yylval=atof(yytext);return NUMBER;}
[\t ]           ;
[\n]            {return 0;}
.                       {return yytext[0];}
%%
int yywrap()
{
        return 0;
}
```

- **4A_A4.y**

```
%{
#include<stdio.h>
#include "y.tab.h"
int yylex();
int yyerror();
%}
%token  NUMBER
%%
arithExp:       exp                     {printf("Ans= %d",$$);}
exp:                    exp             '+'             mulExp
        {$$ = $1 + $3;}
                        |       exp             '-'             mulExp
                {$$ = $1 - $3;}
                        |       mulExp
                        ;
mulExp:                 mulExp          '*'             primary
                {$$ = $1 * $3;}
                        |       mulExp          '/'             primary
                {$$ = $1 / $3;}
                        |       primary
                        ;
primary:                '('     exp ')'                                 {$$
= $2;}
                        |       '-' primary
        {$$ = $2;}
                        |       NUMBER
                        ;
%%
int main()
{
        printf("\nEnter Expression:  ");
        yyparse();
        return 1;
}

int yyerror(char *s)
{
        printf("%s",s);
        return 1;
}
```

# Output

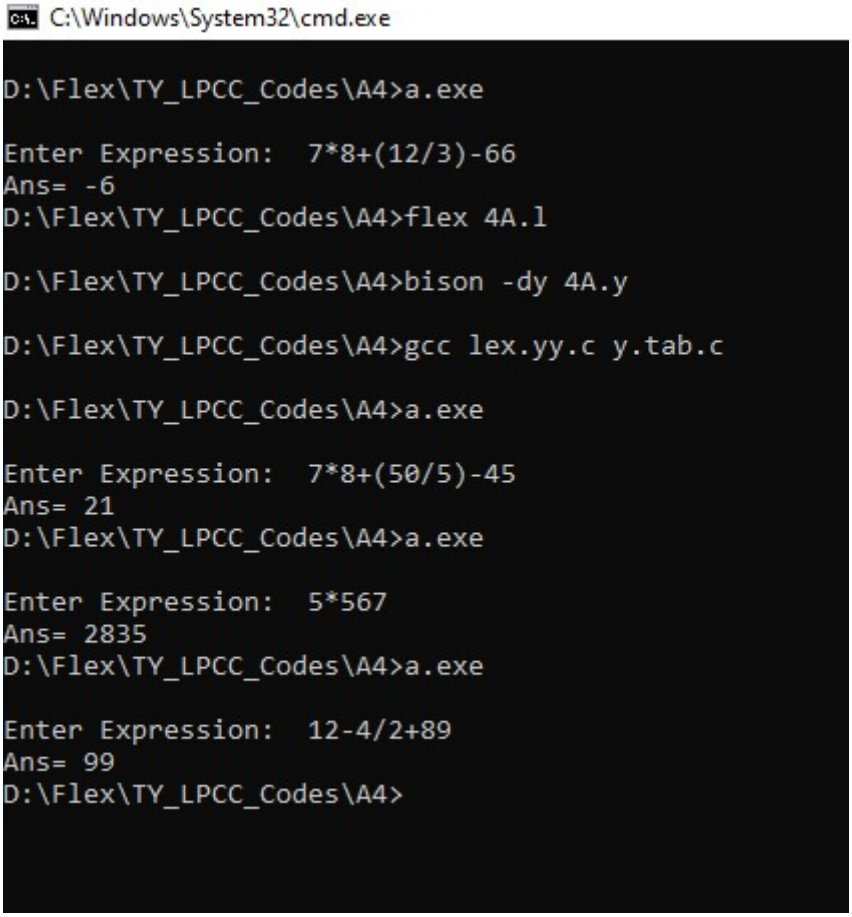
- **Execution Steps :**

  flex 4A_a4.l

  bison -dy 4A_a4.y

  gcc lex.yy.c y.tab.c

  a.exe



# For Floating Point Values

## Code :-

- **4A_AF4.l**

```
%{
#include<stdio.h>
#include "y.tab.h"
%}
%%
(([0-9]*\.[0-9]+)|([0-9]+\.[0-9]*))|([0-9]+)     {yylval.real=(float)
atof(yytext);return NUMBER;}
[\t ]                ;
[\n]                 {return 0;}
.                            {return yytext[0];}
%%
int yywrap()
{
          return 0;
}
```

- **4A_AF4.y**

```
%{
#include<stdio.h>
#include "y.tab.h"
int yylex();
int yyerror();
%}
%union {
float real;
};
%token <real> NUMBER
%type  <real> arithExp
%type   <real> exp
%type  <real>  mulExp
%type  <real>  primary
%%
arithExp:       exp                     {printf("Ans= %f",$$);}
exp:                    exp             '+'             mulExp
        {$$ = $1 + $3;}
                        |       exp             '-'             mulExp
                {$$ = $1 - $3;}
                        |       mulExp
                        ;
mulExp:                 mulExp          '*'             primary
                {$$ = $1 * $3;}
                        |       mulExp          '/'             primary
                {$$ = $1 / $3;}
                        |       primary
                        ;
primary:                '('     exp ')'                                 {$$
= $2;}
                        |       '-' primary
        {$$ = $2;}
                        |       NUMBER
                        ;
%%
int main()
{
        printf("\nEnter Expression:  ");
        yyparse();
        return 1;
}
```

```
int yyerror(char *s)
{
        printf("%s",s);
        return 1;
}
```

# Output

- **Execution Steps :**

  flex 4A_af4.l

  bison -dy 4A_af4.y

  gcc lex.yy.c y.tab.c

  a.exe

```
D:\Flex\TY_LPCC_Codes\A4>flex 4AF.l

D:\Flex\TY_LPCC_Codes\A4>bison -dy 4AF.y

D:\Flex\TY_LPCC_Codes\A4>gcc lex.yy.c y.tab.c

D:\Flex\TY_LPCC_Codes\A4>a.exe

Enter Expression:  7+8/6
Ans= 8.333333
D:\Flex\TY_LPCC_Codes\A4>7*8-(100/3)+9
'7*8-' is not recognized as an internal or external command,
operable program or batch file.

D:\Flex\TY_LPCC_Codes\A4>a.exe

Enter Expression:  7+8/6
Ans= 8.333333
D:\Flex\TY_LPCC_Codes\A4>flex 4AF.l

D:\Flex\TY_LPCC_Codes\A4>bison -dy 4AF.y

D:\Flex\TY_LPCC_Codes\A4>gcc lex.yy.c y.tab.c

D:\Flex\TY_LPCC_Codes\A4>a.exe

Enter Expression:  7/8+6
Ans= 6.875000
D:\Flex\TY_LPCC_Codes\A4>7*8-(100/3)+9
'7*8-' is not recognized as an internal or external command,
operable program or batch file.

D:\Flex\TY_LPCC_Codes\A4>a.exe

Enter Expression:  7*8-(100/3)+9
Ans= 31.666668
D:\Flex\TY_LPCC_Codes\A4>a.exe

Enter Expression:  123/12
Ans= 10.250000
D:\Flex\TY_LPCC_Codes\A4>
```

**Task 2 :=** Evaluate an or check built-in functions using YACC tool.

## Code :-

- **4A_4B.l**

```
%{
        #include<stdio.h>
        #include<math.h>
        #include <string.h>
        #include "y.tab.h"
%}
%%
[ \t]+   ;
sqrt            {return SQRT;}
exp                     {return EXP;}
pow                     {return POW;}
log                     {return LOG;}
strlen          {return STRLEN;}
[0-9]+          {yylval = atoi(yytext);return NUMBER;}
[a-zA-Z][a-zA-Z0-9_]+           {yylval = yytext+1;return STRING;}
.                       {return yytext[0];}
\n              { return 0; }
%%
int yywrap()
{
        return 0;
}
```

- **4A_4B.y**

```
%{
        #include<stdio.h>
        #include<math.h>
        int yylex();
        void yyerror();
%}
%token SQRT EXP      STRLEN NUMBER STRING  LOG POW
%%
stmt : sqroot                   {printf("=%d",$1);return 0;}
                |
          expo                  {printf("=%d",$1);return 0;}
                |
          power                         {printf("=%d",$1);return 0;}
                |
          logs                  {printf("=%d",$1);return 0;}
                |
          strLength             {printf("=%d",$1);return 0;}
          ;
sqroot : SQRT '(' NUMBER ')'            {$$=sqrt($3);}          ;
expo   : EXP '(' NUMBER ')'                     {$$=exp($3);}           ;
logs   : LOG '(' NUMBER ')'                     {$$=log($3);}           ;
power  : POW '(' NUMBER ',' NUMBER ')'
        {$$=pow($3,$5);}                ;
strLength : STRLEN '(' STRING ')'       {$$=strlen($3);}        ;
%%
int main()
{
        yyparse();
        return 1;
}
void yyerror()
{
        printf("This is invalid Statement!\n");
}
```

# Output

- **Execution Steps :**

  flex 4A_4B.l

  bison -dy 4A_4B.y

  gcc lex.yy.c y.tab.c

  a.exe

```
D:\Flex\TY_LPCC_Codes\A4>bison -dy A4_4b.y

D:\Flex\TY_LPCC_Codes\A4>gcc lex.yy.c y.tab.c
A4_4b.l: In function 'yylex':
A4_4b.l:15:9: warning: assignment to 'YYSTYPE' {aka 'int'} from 'c
 [a-zA-Z][a-zA-Z0-9_]+  {yylval = yytext+1;return STRING;}
         ^
A4_4b.y: In function 'yyparse':
A4_4b.y:23:14: warning: implicit declaration of function 'strlen'
 strLength : STRLEN '(' STRING ')' {$$=strlen($3);} ;
              ^~~~~~
A4_4b.y:23:14: warning: incompatible implicit declaration of built
A4_4b.y:23:14: note: include '<string.h>' or provide a declaration
+#include <string.h>
 strLength : STRLEN '(' STRING ')' {$$=strlen($3);} ;
              ^~~~~~
A4_4b.y:23:27: warning: passing argument 1 of 'strlen' makes point
 strLength : STRLEN '(' STRING ')' {$$=strlen($3);} ;
                     ~~~~~~~^~~~~~~~~~~~~~~
A4_4b.y:23:27: note: expected 'const char *' but argument is of ty

D:\Flex\TY_LPCC_Codes\A4>a.exe
strlen(AboliJadhav)
=11
D:\Flex\TY_LPCC_Codes\A4>a.exe
pow(8,3)
=512
D:\Flex\TY_LPCC_Codes\A4>a.exe
log(9)
=2
D:\Flex\TY_LPCC_Codes\A4>a.exe
sqrt(576)
=24
D:\Flex\TY_LPCC_Codes\A4>a.exe
exp(5)
=148
D:\Flex\TY_LPCC_Codes\A4>a.exe
```

**Task 3 :-** To recognize valid variable name using YACC tool.

## Code :-

- **4A_4C.l**

```
%{
#include<stdio.h>
#include "y.tab.h"
%}

%%
new return NEW;
"[" return OPEN_SQ;
"]" return CLOSE_SQ;
"=" return EQ;
"," return COMMA;
"_" return UD;
(["\t"])+ return WS;
[a-zA-Z]+[a-zA-Z0-9_]* return CHAR;
[0-9]+ return DIGIT;
\n return 0;
%%
```

- **4A_4C.y**

```
%{
#include<stdio.h>
#include "y.tab.h"
int yylex();
int yyerror();
%}
%token BUILTIN UD WS CHAR OPEN_SQ CLOSE_SQ EQ NEW SC
COMMA DIGIT
%%
start :  varlist WS varlist  {printf(" NOT Valid Declaration \n");}
      |  varlist UD DIGIT  {printf("Valid Declaration \n");}
      |  varlist  {printf("Valid Declaration \n");}
      |  varlist UD varlist {printf("Valid Declaration \n");}
      ;
varlist : varlist COMMA CHAR | CHAR
                  ;
%%

int yywrap()
{ return 1;
}

int main()
{
        printf("\nEnter variable   :  ");
        yyparse();
        return 1;
}

int yyerror(char *s)
{
        printf("%s",s);
        return 1;
}
```

# Output

- **Execution Steps :**

  flex 4A_4C.l

  bison -dy 4A_4C.y

  gcc lex.yy.c y.tab.c

  a.exe