

Assignment 4 : Write a program to evaluate an arithmetic expression, check built-in functions, and valid variables using YACC specification.

Tools used: Flex and bison

1. To evaluate an arithmetic expression using YACC tool

Source code:

4a.l:

```
%{
#include<stdio.h>
#include "y.tab.h"
extern int yylval;
}%

%%

[0-9]+ {
    yylval=atoi(yytext);
    return NUMBER;
}

[\\t] ;
[\\n] return 0;
. return yytext[0];
%%

int yywrap()
{
return 1;
}
```

4a.y:

```
%{
```

```

#include<stdio.h>

int flag=0;

%}

%token NUMBER

%left '+' '-'
%left '*' '/' '%'
%left '(' ')'

%%

ArithmeticExpression: E{
    printf("\nResult=%d\n", $$);
    return 0;
}

E: E '+' E { $$ = $1 + $3; }
  | E '-' E { $$ = $1 - $3; }
  | E '*' E { $$ = $1 * $3; }
  | E '/' E { $$ = $1 / $3; }
  | E '%' E { $$ = $1 % $3; }
  | '(' E ')' { $$ = $2; }
  | NUMBER { $$ = $1; }

;

%%

void main()
{
    printf("\nEnter Any Arithmetic Expression which can have operations
Addition, Subtraction, Multiplication, Divison, Modulus and Round
brackets:\n");

    yyparse();

    if(flag==0)
        printf("\nEntered arithmetic expression is Valid\n\n");

}

void yyerror()

```

```
{
    printf("\nEntered arithmetic expression is Invalid\n\n");
    flag=1;
}
```

O/P:

```
Enter Any Arithmetic Expression which can have operations Addition, Subtraction, Multiplication, Divison, Modulus and
Round brackets:
3+(5*6)

Result=33

Entered arithmetic expression is Valid
```

```
Enter Any Arithmetic Expression which can have operations Addition, Subtraction, Multiplication, Divison, Modulus and
Round brackets:
4*(4+

Entered arithmetic expression is Invalid
```

2. To evaluate or check built-in functions using YACC tool

Source code:

4b.i

```
%{
#include<stdio.h>
#include "y.tab.h"
extern int yylval;
}%

%%

[0-9]+ {yylval=atoi(yytext); return NUMBER;}
remainder {return REMAINDER;}
sqrt {return Sqrt;}
exp {return EXP;}
[\\t] ;
[\\n] return 0;
. return yytext[0];
%%

int yywrap()
```

```
{
return 1;
}
```

4b.y:

```
%{
#include <stdio.h>
#include <math.h>
int flag=0;
%}

%token REMAINDER SQRT EXP NUMBER

%%

ArithmeticExpression: E{
    printf("\nResult=%d\n", $$);
    return 0;
}

E:REMAINDER('NUMBER','NUMBER')' {if($5==0){printf("nan
error");return;} $$=$3$5;}

| SQRT('NUMBER')' {if($3<0){printf("negative number");return;}
$$=sqrt($3);}

| EXP('NUMBER','NUMBER')' {$$=1;for(int i=0;i<$5;i++) {$$=$*$3;}}

| NUMBER {$$=$1;}

;

%%

void main()
{
    printf("\nEnter a built-in C function:\n");
    yyparse();
    if(flag==0)
        printf("\nEntered function is Valid\n\n");
}

void yyerror()
{
```

```
printf("\nEntered function is Invalid\n\n");  
flag=1;  
}
```

O/P:

```
Enter a built-in C function:  
remainder(4,5)
```

```
Result=4
```

```
Entered function is Valid
```

```
Enter a built-in C function:  
remainder(4,0)  
nan error  
Entered function is Valid
```

```
Enter a built-in C function:  
sqrt(9)
```

```
Result=3
```

```
Entered function is Valid
```

```
Enter a built-in C function:  
sqrt(4,5)
```

```
Entered function is Invalid
```

```
Enter a built-in C function:  
exp(4,5)
```

```
Result=1024
```

```
Entered function is Valid
```

```
Enter a built-in C function:  
exp(4)
```

```
Entered function is Invalid
```

3. To recognize valid variable name using YACC tool.

Source code:

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 ID;
 [0-9]+ return DIGIT;
 \n return 0;
%%
```

4c.y:

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

%token BUILTIN UD WS ID 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 ID | ID;
```

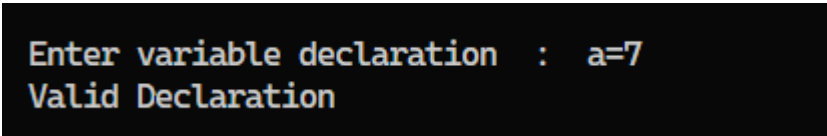
%%

```
int yywrap()
{ return 1;
}

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

int yyerror(char *s)
{
    fprintf(stderr, "\n", s);
    return 1;
}
```

O/P:



```
Enter variable declaration : a=7
Valid Declaration
```

Submitted By:

Shrutika Kailas Hilale

Rollno:321026

PRN:22010744

Batch: A1