**POC Practical’s:**

**2) Count the number of tokens**

%option noyywrap

%{

#include<stdio.h>

int count=0;

%}

%%

"while"|"if"|"else"|"int"|"float" {count++; printf("Keyword: %s\n",yytext);}

[a-zA-Z\_][a-zA-Z0-9]\* {count++; printf("Identifier: %s\n",yytext);}

[{}()|;,] {count++; printf("Separator: %s\n",yytext);}

"+"|"-"|"\*"|"/"|"%"|"=="|"<"|">"|"<="|">=" {count++; printf("Operator: %s\n",yytext);}

[0-9]+ {count++; printf("Integer: %s\n",yytext);}

[0-9]+[.][0-9] {count++; printf("Float: %s\n",yytext);}

. ;

%%

int main()

{

printf("Enter the line: \n");

yylex();

printf("The total numbers of tokens = %d",count);

return 0;

}

1. **Average of Numbers:**

%option noyywrap

%{

#include <stdio.h>

int count = 0;

double sum = 0.0;

%}

%%

[0-9]+ { sum += atof(yytext); count++; }

\n ;

. ;

%%

int main() {

printf("Enter numbers :\n");

yylex();

if (count > 0) {

double average = sum / count;

printf("Average: %.2f\n", average);

} else {

printf("No numbers entered.\n");

}

return 0;

}

**3) Decimal to Octal:**

**4) Frequency of word in file.**

%option noyywrap

%{

#include <stdio.h>

#include <string.h>

int c = 0;

char\* word;

%}

%%

[a-zA-Z]+ {

if (strcasecmp(yytext, word) == 0)

c++;

}

.|\n {}

%%

int main(int argc, char\* argv[]) {

word = "Good";

FILE\* file = fopen("abc.txt", "r");

if (!file) {

printf("Error opening file: abc.txt\n");

return 1;

}

yyin = file;

yylex();

printf("The occurrence of the word '%s' in the file 'abc.txt' is %d\n", word, c);

fclose(file);

return 0;

}

%option noyywrap

%{

#include<stdio.h>

#include<math.h>

%}

%%

[0-9]+ {

int decimal = atoi(yytext);

int octal = 1;

int i = 0;

while (decimal != 0) {

octal += (decimal % 8) \* pow(10, i);

decimal /= 8;

i++;

}

printf("Octal equivalent: %d\n", octal);

}

. ;

%%

int main() {

printf("Enter a decimal number: ");

yylex();

return 0;

}

**5) Check whether it is integer, real number**

**7) Number of lines in File**

%option noyywrap

%{

#include<stdio.h>

int count = 0;

%}

%%

\n { count++; }

. { }

%%

int main(int argc, char \*argv[]) {

FILE \*yyin = fopen("abc.txt", "r+");

yylex();

printf("\nNumber of Lines is %d\n", count);

return 0;

}

**8) Valid C variable**

%option noyywrap

%{

#include<stdio.h>

%}

%%

[a-zA-Z\_][a-zA-Z0-9\_]\* { printf("It is a valid variable name.\n"); }

[0-9][a-zA-Z0-9\_]\* { printf("It is not a valid variable name.\n"); }

.|\n { /\* Ignore any other input \*/ }

%%

int main() {

printf("Enter the variable name: ");

yylex();

return 0;

}

**Or word.**

%option noyywrap

%{

#include<stdio.h>

%}

%%

[0-9]+ { printf("It is an integer\n"); }

[0-9]+"."[0-9]+ { printf("It is a real number\n"); }

[a-zA-Z]+ { printf("It is a word\n"); }

. { printf("Invalid input\n"); }

%%

int main() {

printf("Enter the element: ");

yylex();

return 0;

}

**6) Assign Line number**

%option noyywrap

%{

#include <stdio.h>

int line\_number = 1;

%}

%%

\n { line\_number++; printf("Line %d: ", line\_number);

. { putchar(yytext[0]); } // Print the character

%%

int main() {

printf("Enter text:\n");

printf("Line 1: ");

yylex();

return 0;

}

**9) Positive and Negative number count**

%option noyywrap

%{

#include <stdio.h>

int positive\_count = 0;

int negative\_count = 0;

%}

%%

-?[0-9]+ {

int number = atoi(yytext);

if (number > 0)

positive\_count++;

else if (number < 0)

negative\_count++;

}

. ;

%%

int main() {

printf("Enter numbers (one per line):\n");

yylex();

printf("Positive numbers: %d\n", positive\_count);

printf("Negative numbers: %d\n", negative\_count);

return 0;

}

**10) Calculator using flex**

break;

case 2:

a = a - b;

break;

case 3:

a = a \* b;

break;

case 4:

a = a / b;

break;

case 5:

for (int i = 1; i < b; i++) {

a = pow(a, b);

}

break;

}

op = 0;

}

}

int main(int argc, char \*argv[]) {

yylex();

return 0;

}

int yywrap() {

return 1;

}

%{

#include <stdio.h>



#include <stdlib.h>

#include <math.h>

int op = 0;

float a, b;

void digi();

%}

%%

[0-9]+|[0-9]+"."[0-9]+ { digi(); }

"+" { op = 1; }

"-" { op = 2; }

"\*" { op = 3; }

"/" { op = 4; }

"^" { op = 5; }

\n { printf("\n The Answer: %f\n\n", a); }

. ;

%%

void digi() {

if (op == 0) {

a = atof(yytext);

} else {

b = atof(yytext);

switch (op) {

case 1:

a = a + b;

**11) Calculator using YACC**

int yyerror(const char\* msg) {

printf("Error: %s\n", msg);

return 1;

}

* **The Lex Part: -**

%{

#include "calc.tab.h"

%}

%%

[0-9]+ {

yylval = atoi(yytext);

return NUMBER;

}

[ \t] /\* Skip whitespace \*/

.|\n return \*yytext;

%%

int yywrap() {

return 1;

}

%{

#include <stdio.h>

%}

%token NUMBER

%left '+' '-'

%left '\*' '/'

%left UMINUS

%%

input: /\*empty\*/

| input line

;

line: '\n'

| exp '\n' { printf("Result: %d\n", $1); }

;

exp: NUMBER { $$ = $1; }

| exp '+' exp { $$ = $1 + $3; }

| exp '-' exp { $$ = $1 - $3; }

| exp '\*' exp { $$ = $1 \* $3; }

| exp '/' exp { $$ = $1 / $3; }

| '-' exp %prec UMINUS { $$ = -$2; }

| '(' exp ')' { $$ = $2; }

;

%%

int main() {

yyparse();

return 0;

}