System Software

Nehal Jhajharia

Tutorial 2

1. Write a Lex program to compute area of a circle.

Code:

```
%{
 #include <stdio.h> #include <stdlib.h> int r;
 %}


%%
 [0-9]+ {r = atoi(yytext); printf("area: %f\n", 3.14 * r * r);} %%


int main() { yylex();


return 0; }
```

2. Write a Lex program to calculate the simple interest.

Code:

```
%{
 #include <stdio.h> #include <stdlib.h> float p, r, t;
 %}


%%
 [0-9]+\.?[0-9]* {return atof(yytext);} %%


int main() {
 printf("Enter prinicipal amount: ");
 p = yylex();
 printf("Enter rate of interest: ");
 r = yylex();
 printf("Enter time: ");
 t = yylex();
 printf("Simple interest: %f\n", p * r * t); return 0;


}
```

3. Write a Lex program that convert Fahrenheit to Celsius.

Code:

```
%{
 #include <stdio.h> #include <stdlib.h> %}


%%
 \-?[0-9]+\.?[0-9]* return atof(yytext); %%


int main() {
 printf("Enter temperature in Farenheit: "); float f = yylex();
 float c = (f - 32.0) * 5.0 / 9.0; printf("Temperature in Celsius: %f\n", c); return 0;


}
```

4. Write a Lex program to swap two number with and without

using temporary variable.

Code:

```
%{
 #include <stdio.h> #include <stdlib.h> %}


%%
 [0-9]+ {return atoi(yytext);} %%


void swapWithThird(int *n1, int *n2) { int temp;


temp = *n1;
 *n1 = *n2;
 *n2 = temp;
 printf("Numbers: %d and %d\n", *n1, *n2);


}
 void swapWithoutThird(int *n1, int *n2) {
```

```
*n1 += *n2;
 *n2 = *n1 - *n2;
 *n1 -= *n2;
 printf("Numbers: %d and %d\n", *n1, *n2);


}


int main() { int n1, n2;


printf("Enter first number: \n");
 n1 = yylex();
 printf("Enter second number: \n"); n2 = yylex();


swapWithThird(&n1, &n2); swapWithoutThird(&n1, &n2); return 0;


}
```

5. Write a Lex program that read two number and performs their division. If the division is not possible, then an error message, “Division not possible” is displayed.

Code:

```
%{
 #include <stdio.h> #include <stdlib.h> %}

%%
 \-?[0-9]+\.?[0-9]* return atof(yytext); %%

int main() {
 float f1, f2;
 printf("Enter first number: ");
 f1 = yylex();
 printf("Enter second number: "); f2 = yylex();
 if(f2 != 0) {

printf("Division: %f\n", f1 / f2); } else {

printf("Division by zero error."); }

return 0; }
```

6. Write a Lex program to recognize valid arithmetic expression

and identify the identifiers and operators.

Code:

```
%{
 #include <stdio.h>
 char op;
 int pos = 1;
 int idFound = 0;
 %}
 %%
 [a-zA-Z]* { if(idFound == 0) {

printf("Identifier = %s\n", yytext);

idFound = 1; } else {

printf("Invalid\n");
```

```
exit(0); }}

[-+*/] { op = yytext[0]; if(idFound == 1) {

printf("Operator = %c\n", op);

idFound = 0; } else {

printf("Invalid\n");

exit(0); }}

%%

int main() { yylex(); printf("Valid\n"); return 0;
```

}

7. Write a Lex program to count the Positive numbers, Negative numbers and Fractions.

Code:

```
%{
 #include <stdio.h>
 int posInt = 0, negInt = 0, posFrac = 0, negFrac = 0; %}


%%
 [0-9]+ posInt++; [-][0-9]+ negInt++; [0-9]*\.[0-9]* posFrac++; [-][0-9]*\.[0-9]* negFrac++;
%%


int main() { yylex();


printf("Positive integers count = %d, Negative integers count = %d\n Positive fractions
count = %d, Negative fractions count = %d\n", \
```

posInt, negInt, posFrac, negFrac); }

8. Write a Lex program to count the number of words.

Code:

```
%{
 #include <stdio.h> #include <stdlib.h> int wordCount = 0; %}


%%
 [a-zA-Z]+ wordCount++; %%


int main() { yylex();


printf("Word count: %d\n", wordCount);


return 0; }
```