

**LAB DAY 02 Answers**

1. **Write a LEX program to find the length of the longest word.**

% {

int counter = 0; %

}

%

% [a - zA - Z] + {

if (yyleng > counter) {

counter = yyleng;

}

} %

%

main() {

yylex();

printf("largest: %d", counter);

printf("\n");

}

1. A networking company wants to validate the URL for their clients. Write a LEX program to implement the same.

%%

((http)|(ftp))s?:\/\/[a-zA-Z0-9](.[a-z])+(.[a-zA-Z0-9+=?]\*)\* {printf("\nURL Valid\n");}

.+ {printf("\nURL Invalid\n");}

%%

void main()

{

printf("\nEnter URL : ");

yylex();

printf("\n");

}

int yywrap()

{

}

1. School management wants to validate DOB of all students. Write a LEX program to implement it.

%{

#include <stdio.h>

%}

%%

[0-9]{1,2}-[0-9]{1,2}-[0-9]{4} { printf("Valid date of birth\n"); }

.+ { printf("Invalid date of birth\n"); }

%%

int yywrap(void){}

int main() {

yylex();

return 0;

}

1. Write a LEX program to recognize a word and relational operator.

%{

#include <stdio.h>

#include <string.h>

    int operators\_count = 0, operands\_count = 0, valid = 1, top = -1, l = 0, j = 0;

    char operands[10][10], operators[10][10], stack[100];

%}

%%

"(" {

    top++;

    stack[top] = '(';

}

"{" {

    top++;

    stack[top] = '{';

}

"[" {

    top++;

    stack[top] = '[';

}

")" {

    if (stack[top] != '(') {

        valid = 0;

    }

    else if(operands\_count>0 && (operands\_count-operators\_count)!=1){

        valid=0;

    }

    else{

        top--;

        operands\_count=1;

        operators\_count=0;

    }

}

"}" {

    if (stack[top] != '{') {

        valid = 0;

    }

    else if(operands\_count>0 && (operands\_count-operators\_count)!=1){

        valid=0;

    }

    else{

        top--;

        operands\_count=1;

        operators\_count=0;

    }

}

"]" {

    if (stack[top] != '[') {

        valid = 0;

    }

    else if(operands\_count>0 && (operands\_count-operators\_count)!=1){

        valid=0;

    }

    else{

        top--;

        operands\_count=1;

        operators\_count=0;

    }

}

"+"|"-"|"\*"|"/" {

    operators\_count++;

    strcpy(operators[l], yytext);

    l++;

}

[0-9]+|[a-zA-Z][a-zA-Z0-9\_]\* {

    operands\_count++;

    strcpy(operands[j], yytext);

    j++;

}

%%

int yywrap()

{

    return 1;

}

int main()

{

    int k;

    printf("Enter the arithmetic expression: ");

    yylex();

    if (valid == 1 && top == -1) {

        printf("\nValid Expression\n");

    }

    else

        printf("\nInvalid Expression\n");

    return 0;

}

1. Write a LEX code to replace a word with another word in a file.

%{

#include<stdio.h>

%}

%%

"greeks" {printf("replacementword");}

.|\n {putchar(yytext[0]);}

%%

int yywrap(void){}

int main(){

yylex();

return 0;

}

1. A School student was asked to do basic mathematical operations. Implement a LEX program to implement the same.

%{

#include<stdio.h>

float op1=6,op2=7;

%}

%%

"+" {printf("sum =%lf",op1+op2);}

"-" {printf("diff=%lf",op1-op2);}

"\*" {printf("mul=%lf",op1\*op2);}

"/" {printf("div=%lf",op1/op2);}

. {printf("enter proper operator.");}

%%

int yywrap(){}

int main()

{

printf("Enter the Operator::");

yylex();

}

1. Write a LEX Program to check the email address is valid or not.

%{

int flag=0;

%}

%%

[a-z . 0-9]+@[a-z]+".com"|".in" { flag=1; }

%%

int main()

{

yylex();

if(flag==1)

printf("Accepted");

else

printf("Not Accepted");

}

int yywrap()

{ return 1;

}

1. Write a LEX Program to convert the substring abc to ABC from the given input string.

%{

#include

#include

int i;

%}

%%

[a-z A-Z]\* {

for(i=0;i<=yyleng;i++)

{

if((yytext[i]=='a')&&(yytext[i+1]=='b')&&(yytext[i+2]=='c'))

{

yytext[i]='A';

yytext[i+1]='B';

yytext[i+2]='C';

}

}

printf("%s",yytext);

}

[\t]\* return;

.\* {ECHO;}

\n {printf("%s",yytext);}

%%

main()

{

yylex();

}

int yywrap()

{

return 1;

}