

**Exp No: 4**

**Date:**

## **DESIGN A DESK CALCULATOR USING LEX TOOL**

### **AIM:**

To create a calculator that performs addition, subtraction, multiplication and division using lex tool.

### **ALGORITHM:**

1. Initialize variables and declare a function prototype.
2. Define patterns for digits, arithmetic operations, and line breaks.
3. Implement lexical rules to perform actions based on matched patterns.
4. Define a function to convert tokens to floats and perform arithmetic operations.
5. Invoke lexical analysis in the main function.
6. Indicate the end of input with the yywrap() function.

**PROGRAM:**

```
% {
int op = 0,i;
float a, b;
int digi();
% }

dig [0-9]+|([0-9]*)"."([0-9]+)
add "+"
sub "-"
mul "*"
div "/"
pow "^"
ln \n

%%

{ dig } { digi(); }
{ add } { op=1; }
{ sub } { op=2; }
{ mul } { op=3; }
{ div } { op=4; }
{ pow } { op=5; }
{ ln } { printf("\n The Answer :%f\n\n",a); }

%%

int digi()
{
if(op==0)

/* atof() is used to convert
- the ASCII input to float */
a=atof(yytext);

else
{
b=atof(yytext);

switch(op)
{
case 1:a=a+b;
break;

case 2:a=a-b;
break;

case 3:a=a*b;
break;
```

```

case 4:a=a/b;
break;

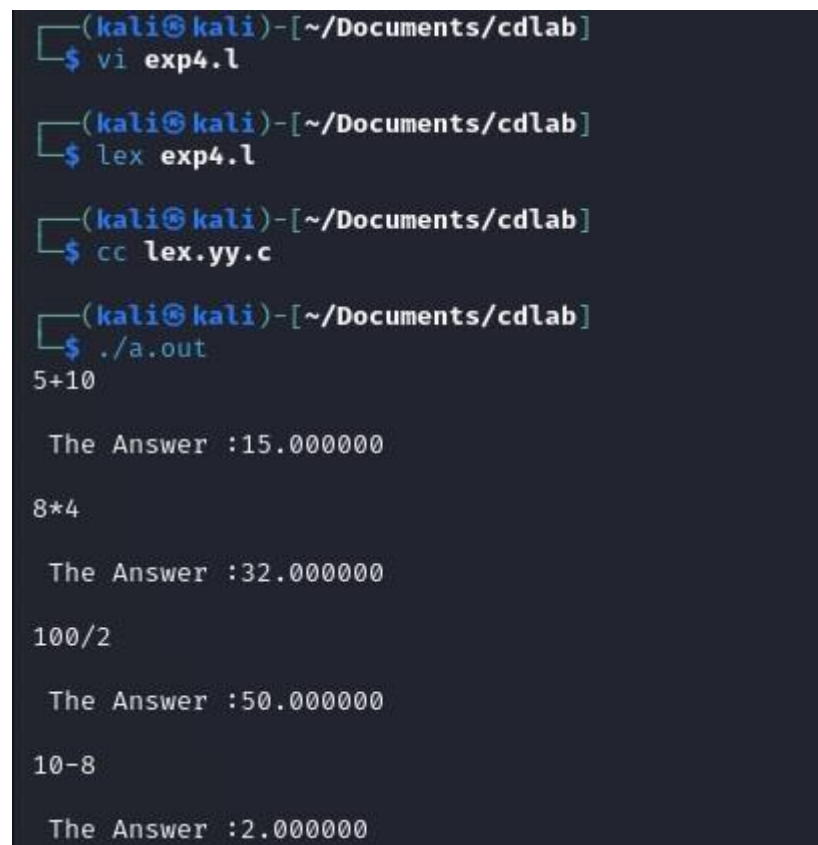
case 5:for(i=a;b>1;b--)
a=a*i;
break;
}
op=0;
}
}

int main(int argv,char *argc[])
{
yylex();
}

int yywrap()
{
return 1;
}

```

#### OUTPUT:



```

(kali㉿kali)-[~/Documents/cdlab]
$ vi exp4.l

(kali㉿kali)-[~/Documents/cdlab]
$ lex exp4.l

(kali㉿kali)-[~/Documents/cdlab]
$ cc lex.yy.c

(kali㉿kali)-[~/Documents/cdlab]
$ ./a.out
5+10

The Answer :15.000000

8*4

The Answer :32.000000

100/2

The Answer :50.000000

10-8

The Answer :2.000000

```

#### RESULT:

Thus, a calculator that performs addition, subtraction, multiplication and division using lex tool is implemented.

Roll Number: 210701072  
Name: S. Harinisubashree