DIT UNIVERSITY DEHRADUN

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Lab Manual for the Academic Year 2017-18

Subject : Compiler Design Lab
Subject code : CS304
Course coordinator : Dr. Garima Verma
HOD : Prof Vishal Bharti

Prepared by:	Reviewed by:	Approved by:
Dr. Garima Verma		

List of Experiments

1. Write a program in C to count number of spaces in a line.
2. Write a program in C to count number of characters, spaces and digits in a line.
3. Write a C program to identify whether a given line is a comment or not.
4. Write a C program to recognize strings under 'a*' , 'abb', and a*b+ .
5. Write a C program to test whether a given identifier is valid or not**.**
6. Write a C program to test whether a given operator is valid or not. (Logical, arithmetic**).**
7. Install Flex for windows. Write a program to print whether the word is a collection of lowercase or upper case
   e.g. Garima

   O/p: Upper lower lower lower lower lower

8. Write a program using Lex to print any arithmetic expression in the form of tokens
   E.g. 2 + 4 * 3

   O/p – Number plus/op Number Multi/op Number

9. Write a program using Lex to print any arithmetic expression in the form of tokens
   E.g. a= b +c

   O/p – identifier plus/op identifier

10. Write a program in Lex to identify whether letter is consonant or vowel. E.g. gari
    O/P – consonant vowel consonant vowel

11. Design a simple calculator using Lex and Yacc.

GENERAL INSTRUCTIONS FOR LABORATORY CLASSES:-

DO'S

(1) Without Prior permission do not enter into the Laboratory.

(2) While entering into the LAB students should wear their ID cards.

(3) The Students should come with proper uniform.

(4) Students should come with observation and record note book to the laboratory.

(5) Students should maintain silence inside the laboratory.

(6) After completing the laboratory exercise, make sure to shutdown the system properly.

DONT'S

(1)Students bringing the bags inside the laboratory.

(2)Students wearing slippers/shoes insides the laboratory.

(3)Students using the computers in an improper way.

(4)Students bringing pen drive or other secondary storage device inside the laboratory.

(5)Students using mobile phones inside the laboratory.

(6)Students making noise inside the laboratory.

Pre-Requisite

The student should have knowledge of concepts of Theory of Automata. He or She should be well verse with C language.

Software Requirements

1. C or C++
2. Flex for windows

Lab Program-1

Write a program in C to count number of spaces in a line

PROGRAM

```
#include <stdio.h>
int main() {
   char line[150];
   int space,i;
     space = 0;
   printf("Enter a line of string: ");
   fgets(line, sizeof(line), stdin);

   for (i = 0; line[i] != '\0'; ++i) {
      if (line[i] == ' ') {
         ++space;
      }
   }

 printf("\nWhite spaces: %d", space);
   return 0;
}
```

OUTPUT

```
Enter a line of string: this is dit university

White spaces: 3
```

Lab Program-2

Write a program in C to count number of spaces, characters and digits in a line.

PROGRAM

```
#include <stdio.h>
int main() {
   char line[150];
   int characters, digit, space;

   characters =  digit = space = 0;

   printf("Enter a line of string: ");
   fgets(line, sizeof(line), stdin);

   for (i = 0; line[i] != '\0'; ++i) {
      if ((line[i] >= 'a' && line[i] <= 'z') || (line[i] >= 'A' && line[i] <= 'Z')) {
         ++ characters;
      } else if (line[i] >= '0' && line[i] <= '9') {
         ++digit;
      } else if (line[i] == ' ') {
         ++space;
      }
   }

   printf("Characters : %d", characters);
   printf("\nDigits : %d", digit);
   printf("\nWhite spaces : %d", space);
   return 0;
}
```

OUTPUT

```
Enter a line of string: this is 123
Characters : 6
Digits : 3
White spaces : 2_
```

Lab Program-3

Write a program in C to identify whether a given line is comment or not.

PROGRAM

```
#include<stdio.h>
#include<conio.h>
void main()
{
char com[30];int i=2,a=0;
clrscr();
printf("\n Enter comment:");
gets(com);
if(com[0]=='/')
{
if(com[1]=='/')
{
        printf("\n It is a comment");
        goto LABEL;
}
else if(com[1]=='*')
{
        for(i=2;i<=30;i++)
        {
                if (com[i]=='*'&&com[i+1]=='/')
                {
                        printf ("\n It is a comment");
                        a=1;
                        break;
                }
                else
                        continue;
        }
}
if(a==0)
        printf("\n It is not a comment");
}
LABEL:
getch();
```

```
}
```

OUTPUT

```
Enter comment:/*this is test */

It is a comment
```

```
Enter comment:/sdssdff

It is not a comment
```

Lab Program-4

Write a C program to recognize strings under 'a*' ,'abb', and a*b+.

PROGRAM

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<stdlib.h>
void main()
{
char s[20],c;
int state=0,i=0;
clrscr();
printf("\n Enter a string:");
gets(s);
while(s[i]!='\0')
{

switch(state)
{
case 0:
        c=s[i++];
        if(c=='a')
                state=1;
        else if(c=='b')
                state=2;
        else
                state=6;
        break;
case 1:
        c=s[i++];
        if(c=='a')
                state=3;
        else if(c=='b')
                state=4;
        else
                state=6;
        break;
case 2:
        c=s[i++];
        if(c=='a')
                state=6;
        else if(c=='b')
                state=2;
        else
```

```
                state=6;
        break;
case 3:
        c=s[i++];
        if(c=='a')
                state=3;
        else if(c=='b')
                state=2;
        else
                state=6;
        break;
case 4:
        c=s[i++];
        state=6;
        if(c=='a')
                state=6;
        else if(c=='b')
                state=5;
        else
                state=6;
        break;
case 5:
        c=s[i++];
        if(c=='a')
                state=6;
        else if(c=='b')
                state=2;
        else
                state=6;
        break;
case 6:
        printf("\n %s is not recognized.",s);
        exit(0);
}
}

if(state==1)
printf("\n %s is accepted under rule 'a'",s);
else if((state==2)||(state==4))
printf("\n %s is accepted under rule 'a*b+'",s);
else if(state==5)
printf("\n %s is accepted under rule 'abb'",s);
else
        printf("\n String is not accepted");
getch();
}
```

OUTPUT

```
Enter a string:abbbb

abbbb is accepted under rule 'a*b+'
```

Lab Program-5

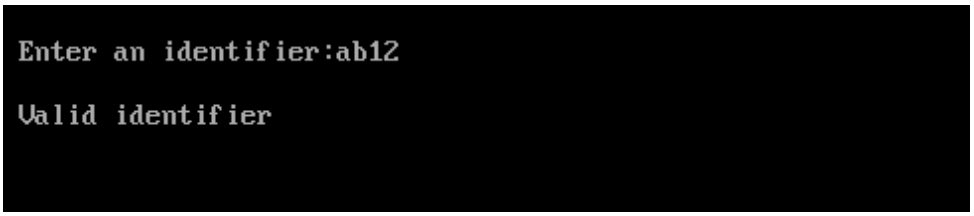
Write a C program to test whether a given identifier is valid or not.

PROGRAM

```
#include<stdio.h>
#include <conio.h>
#include<conio.h>
#include<ctype.h>
void main()
{
char a[10];
int flag;
int i=1;
clrscr();
printf("\n Enter an identifier:");
gets(a);
if(isalpha(a[0]))
        flag=1;
else
        printf("\n Not a valid identifier");
while(a[i]!='\0')
{
        if(!isdigit(a[i])&&!isalpha(a[i]))
        {
                flag=0;
                break;
        }
i++;
}
if(flag==1)
        printf("\n Valid identifier");
getch();
}
```

OUTPUT

```
Enter an identifier:34ws

Not a valid identifier
```


Enter an identifier:ab12
Valid identifier

Lab Program-6

Write a C program to test whether a given operator is valid or not. (Logical, arithmetic)

PROGRAM

```
#include<stdio.h>
#include<conio.h>
void main()
{
char s[5];
clrscr();
printf("\n Enter any operator:");
gets(s);
switch(s[0])
{
case'&':
        if(s[1]=='&')
                printf("\nLogical AND");
        else
                printf("\n Bitwise AND");
        break;
case'|':
        if(s[1]=='|')
                printf("\nLogical OR");
        else
                printf("\nBitwise OR");
        break;
case'+':
        printf("\n Addition");
        break ;
case'-':
        printf("\nSubstraction");
        break;
case'*':
        printf("\nMultiplication");
        break;
case'/':
        printf("\nDivision");
        break;
case'%':
        printf("Modulus");
```

```
        break;
default:
        printf("\n Not a operator");
}
getch();

}
```

OUTPUT

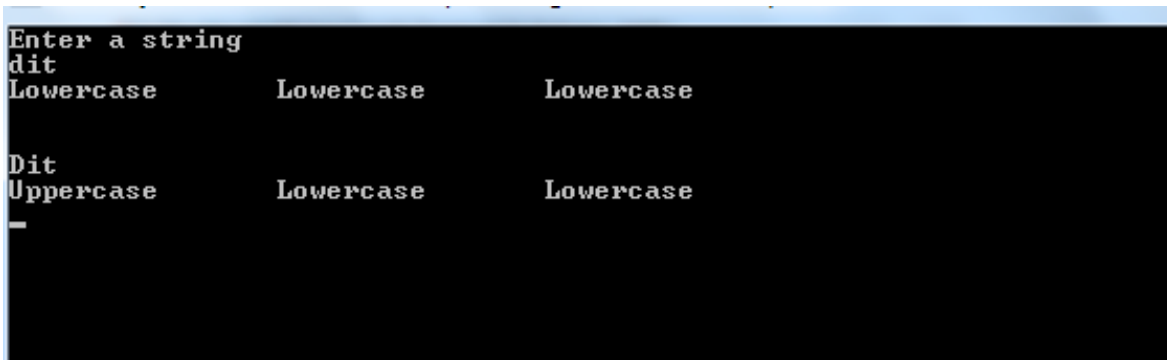
Lab Program-7

Install Flex for windows. Write a program to print whether the word is a collection of lowercase or upper case.

PROGRAM

```
%{

#include<stdio.h>
int Upper=0;
int Lower=0;
%}

%%
[A-Z] {printf("Uppercase\t");Upper++;}
[a-z] {printf("Lowercase\t");Lower++;}
%%

int yywrap()
{
return 1;
}

main()
{
printf("Enter a string\n");
yylex();

printf("Uppercase=%d and Lowercase=%d",Upper,Lower);
}
```

OUTPUT



Lab Program-8

Write a program using Lex to print any arithmetic expression in the form of tokens.

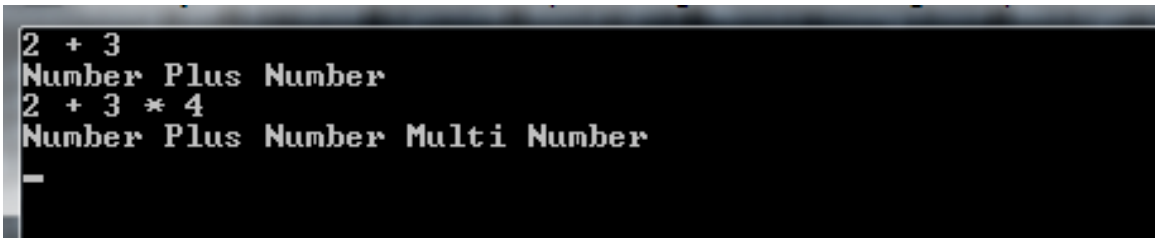
E.g. 2 + 4 * 3

O/p – Number plus/op Number Multi/op Number

PROGRAM

```
%option noyywrap
%{
	#include<stdio.h>
%}

%%
"+" {printf("Plus");}
"-" {printf("Minus");}
"*" {printf("Multi");}
"/" {printf("Divide");}
[0-9]* {printf("Number");}
%%
int main(int agrc, char **argv)
{

	yylex();
}
```

OUTPUT



Lab Program-9

Write a program using Lex to print any arithmetic expression in the form of tokens.

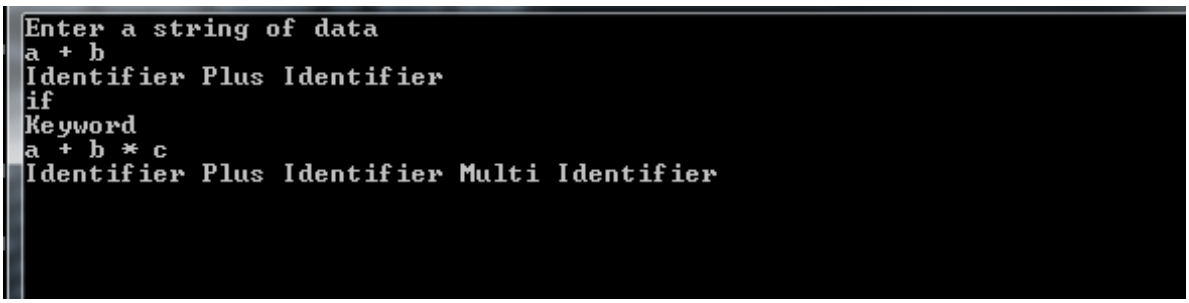
E.g. a= b +c

O/p – identifier plus/op identifier

PROGRAM

```
%{
#include<stdio.h>
%}
%%
"if"|"else"|"while"|"do"|"switch"|"case" {printf("Keyword");}
[a-zA-Z][a-z|0-9]* {printf("Identifier");}
[0-9]* {printf("Number");}
"+" {printf("Plus");}
"-" {printf("Minus");}
"*" {printf("Multi");}
"/" {printf("Divide");}
"!"|"@"|"&"|"^"|"%"|"$"|"#" {printf("Special Character");}
%%
int yywrap()
{
return 1;
}
main()
{
printf("Enter a string of data\n");
yylex();
}
```

OUTPUT



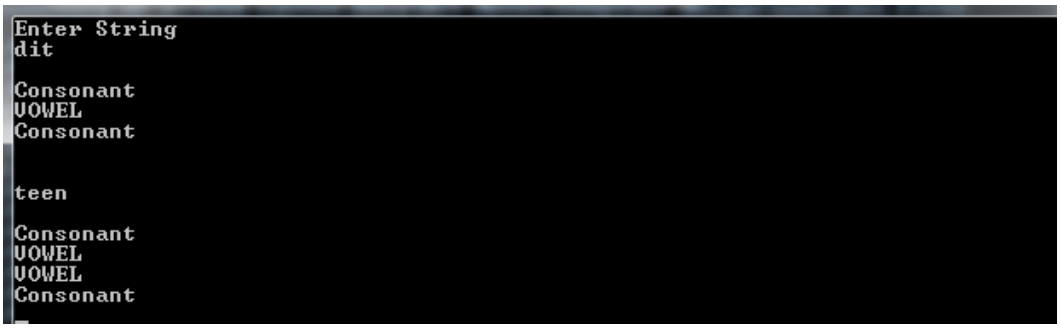
Lab Program-10

Write a program in Lex to identify whether letter is consonant or vowel.

PROGRAM

```
%option noyywrap
%{
#include<stdio.h>
int vowel=0;
int cons=0;
%}
%%
"a"|"e"|"i"|"o"|"u"|"A"|"E"|"I"|"O"|"U" {printf("\nVOWEL");vowel++;}
[a-zA-z] {printf("\nConsonant");cons++;}

%%

main(int argc, char **argv)
{
printf("Enter String \n");
yylex();
return 0;
}
```

OUTPUT



Lab Program-11

Design a simple calculator using Lex and Yacc.

PROGRAM

LEX PART:

```
%{
#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;
}
```

YACC PART:

```
%{
          #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 Arithmetic Expression using +, -, *, /, Mod 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;
}
```

OUTPUT