CD LAB PROGRAMS

1. LEX Program to count the number of vowels and consonants in a given string.

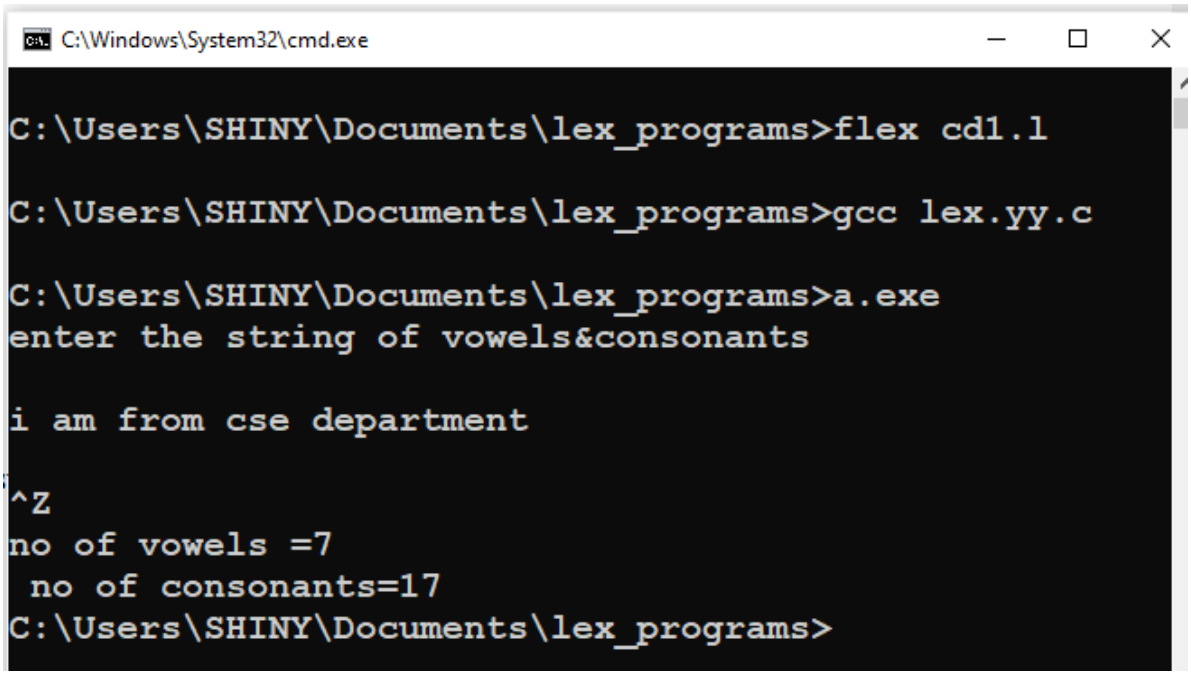
Program:

cd1.l - Notepad

File Edit Format View Help

```
/*   1. LEX Program to count the number of vowels and consonants in a given string.*/
%{
        int vow_count=0;
        int const_count=0;
%}
%%
[aeiouAEIOU] {vow_count++;}
[a-z A-Z] {const_count++;}
%%
int yywrap(){}
int main()
{
        printf("enter the string of vowels&consonants");
        yylex();
        printf("no of vowels =%d \n no of consonants=%d",vow_count,const_count);
        return 0;
}
```

Output:



2. **LEX Program to count the number of lines, words, and characters in the input.**

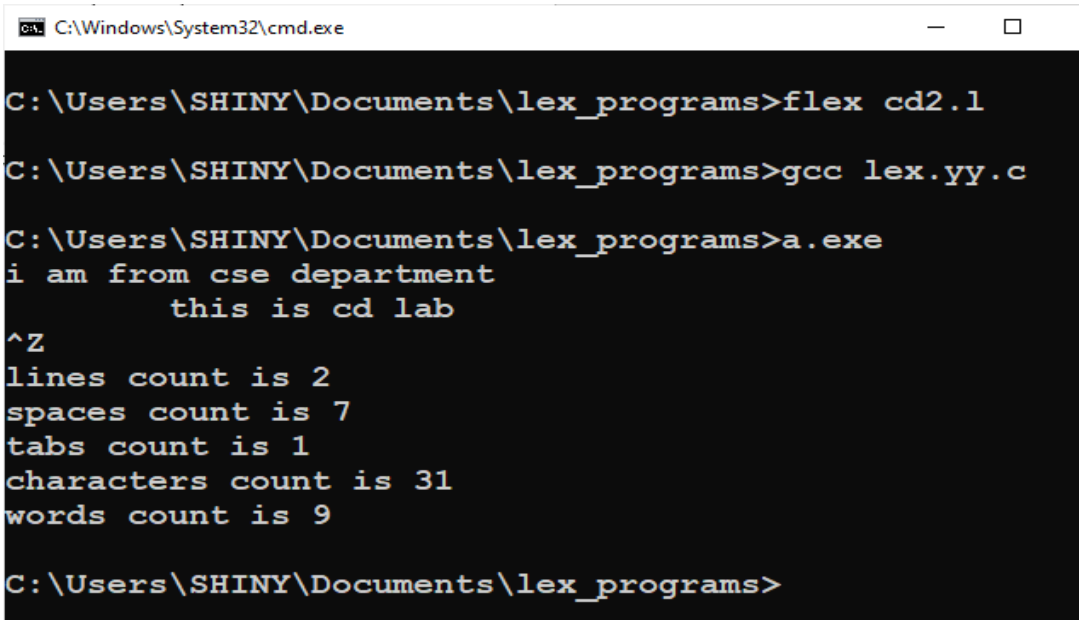
Program:

cd2.l - Notepad

File Edit Format View Help

```
/*2. LEX Program to count the number of lines, words, and characters in the input.*/
%{
        int lc=0,sc=0, tc=0,ch=0, c=0, wc=0;
%}
%%
\n {lc++;}
([ ])+ {sc++;}
\t {tc++;}
. {ch++;c++;}
[a-zA-Z0-9]+ {wc++;c=c+yyleng;}
%%
int yywrap(){}
int main()
{
        yylex();
        printf("lines count is %d\n",lc);
        printf("spaces count is %d\n",sc);
        printf("tabs count is %d\n",tc);
        printf("characters count is %d\n",c);
        printf("words count is %d\n",(wc+ch));
}
```

Output:



3. **LEX Program to count the number of integers and floating point numbers appearing in the input.**

Program:

cd3.l - Notepad

File Edit Format View Help

```
/*   3. LEX Program to count the number of integers and floating point numbers appearing in the input.*/
%{
#include<stdio.h>
int numofint=0,numoffloat=0;
%}
%%
[0-9]+"."[0-9]+ {ECHO; printf("\nDecimal Number\n");numoffloat++;}

[0-9]+ {ECHO; printf("\nInteger Number\n");numofint++;}
%%
int yywrap()
{return 1;}
int main()
{
yylex();
printf("Number of Decimal Numbers:%d\nnumber of Integer Numbers:%d",numoffloat,numofint);
return 0;
}
```

Output:

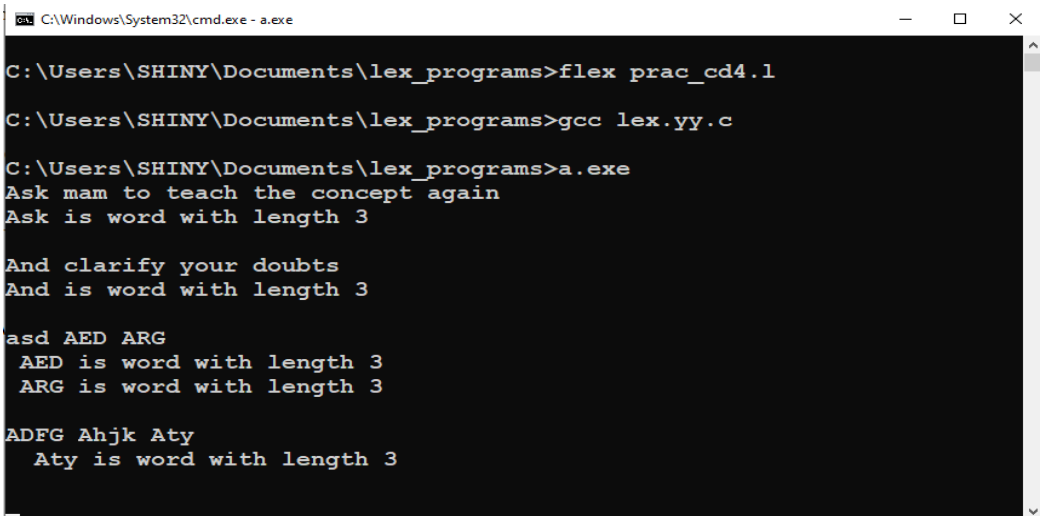
```
C:\Users\SHINY\Documents\lex_programs>a.exe
12 2.3 3.4 5.6 78 98
12
Integer Number
 2.3
Decimal Number
 3.4
Decimal Number
 5.6
Decimal Number
 78
Integer Number
 98
Integer Number

^Z
Number of Decimal Numbers:3
number of Integer Numbers:3
C:\Users\SHINY\Documents\lex_programs>
```

4. LEX Program to list out all words of length three, starting with "A".

Program:

```
/*4. LEX Program to list out all words of length three, starting with "A" .*/
%{
#include<stdio.h>
int length;
%}
%%
[a-zA-Z0-9]+ {length=yyleng;fun(length);}
%%
fun(int l)
{
        if(yytext[0]=='A'){
                if(length==3){
                        printf("%s is word with length 3",yytext);
                }
                /*else{
                        printf("word is not of length 3");
                }*/
        }
        /*else{
        printf("word is not starting with A");
        }*/
}
int yywrap(){}
int main()
{
yylex();
return 0;
}
```

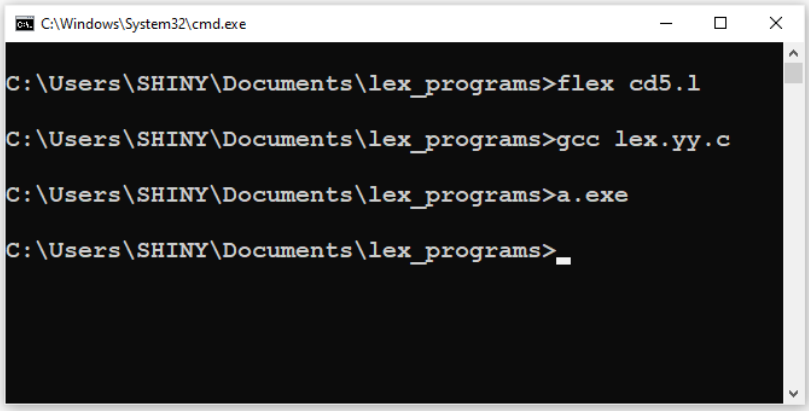
Output:



5. **LEX Program to list out all C-like comments (both single line and multi line comments) from a text file.**

Program:

```
/*5. LEX Program to list out all C-like comments (both single line and multi line comments) from a text file.*/
%{
#include<stdio.h>
#include<stdlib.h>
%}
%%
\/\/(.*) {};
\/\*(.*\n)*.*\*\/ {};
%%
int yywrap()
{
return 1;
}
int main()
{
yyin=fopen("input6.c","r");
yyout=fopen("out.c","w");
yylex();
return 0;
}
```



input6.c

```
1  /*Program to print welcome message*/
2  void main()
3  {
4  //declaration
5  printf("welcome");
6  //End
7  }
```

out.c

```
1
2  void main()
3  {
4
5  printf("welcome");
6
7  }
```

6. **Write a C program to find FIRST.**

Program to find First:

first.c folow.c

```c
#include<stdio.h>
#include<ctype.h>
void FIRST(char[],char );
void addToResultSet(char[],char);
int numOfProductions;
char productionSet[10][10];
main()
{
    int i;
    char choice;
    char c;
    char result[20];
    printf("How many number of productions ? :");
    scanf(" %d",&numOfProductions);
    for(i=0;i<numOfProductions;i++)//read production string eg: E=E+T
    {
        printf("Enter productions Number %d : ",i+1);
        scanf(" %s",productionSet[i]);
    }
    do
    {
        printf("\n Find the FIRST of  :");
        scanf(" %c",&c);
        FIRST(result,c); //Compute FIRST; Get Answer in 'result' array
        printf("\n FIRST(%c)= { ",c);
        for(i=0;result[i]!='\0';i++)
```

first.c folow.c

```c
        for(i=0;result[i]!='\0';i++)
        printf(" %c ",result[i]);       //Display result
        printf("}\n");
         printf("press 'y' to continue : ");
        scanf(" %c",&choice);
    }
    while(choice=='y'||choice =='Y');
}
/*
 *Function FIRST:
 *Compute the elements in FIRST(c) and write them
 *in Result Array.
 */
void FIRST(char* Result,char c)
{
    int i,j,k;
    char subResult[20];
    int foundEpsilon;
    subResult[0]='\0';
    Result[0]='\0';
    //If X is terminal, FIRST(X) = {X}.
    if(!(isupper(c)))
    {
        addToResultSet(Result,c);
            return ;
    }
```

first.c folow.c

```
52      //If X is non terminal
53      //Read each production
54      for(i=0;i<numOfProductions;i++)
55      {
56  //Find production with X as LHS
57          if(productionSet[i][0]==c)
58          {
59  //If X ? e is a production, then add e to FIRST(X).
60   if(productionSet[i][2]=='$') addToResultSet(Result,'$');
61              //If X is a non-terminal, and X ? Y1 Y2 … Yk
62              //is a production, then add a to FIRST(X)
63              //if for some i, a is in FIRST(Yi),
64              //and e is in all of FIRST(Y1), …, FIRST(Yi-1).
65        else
66              {
67                  j=2;
68                  while(productionSet[i][j]!='\0')
69                  {
70                  foundEpsilon=0;
71                  FIRST(subResult,productionSet[i][j]);
72                  for(k=0;subResult[k]!='\0';k++)
73                      addToResultSet(Result,subResult[k]);
74                   for(k=0;subResult[k]!='\0';k++)
75                      if(subResult[k]=='$')
76                      {
77                          foundEpsilon=1;
```

first.c folow.c

```
78                                  break;
79                              }
80                          //No e found, no need to check next element
81                          if(!foundEpsilon)
82                              break;
83                          j++;
84                      }
85                  }
86              }
87          }
88              return ;
89          }
90          /* addToResultSet adds the computed
91           *element to result set.
92           *This code avoids multiple inclusion of elements
93            */
94          void addToResultSet(char Result[],char val)
95          {
96              int k;
97              for(k=0 ;Result[k]!='\0';k++)
98                  if(Result[k]==val)
99                      return;
100             Result[k]=val;
101             Result[k+1]='\0';
102         }
103
```

Output:

```
C:\Users\SHINY\Documents\lex_programs\first.exe
Enter productions Number 5 : B-$

 Find the FIRST of  :A

 FIRST(A)= {  c  $ }
press 'y' to continue : y

 Find the FIRST of  :S

 FIRST(S)= {  a }
press 'y' to continue : y

 Find the FIRST of  :B

 FIRST(B)= {  d  $ }
press 'y' to continue : n

--------------------------------
Process exited after 173.2 seconds with return value 110
Press any key to continue . . .
```

7. **Write a C program to find FOLLOW.**

Program to find Follow:

first.c folow.c

```c
#include<stdio.h>
#include<string.h>
int n,m=0,p,i=0,j=0;
char a[10][10],followResult[10];
void follow(char c);
void first(char c);
void addToResult(char);
int main()
{
 int i;
 int choice;
 char c,ch;
 printf("Enter the no.of productions: ");
scanf("%d", &n);
 printf(" Enter %d productions\nProduction with multiple terms should be give as separate productions \n", n);
 for(i=0;i<n;i++)
  scanf("%s",a[i]);
    // gets(a[i]);
 do
 {
  m=0;
  printf("Find FOLLOW of -->");
  scanf(" %c",&c);
  follow(c);
  printf("FOLLOW(%c) = { ",c);
  for(i=0;i<m;i++)
```

first.c folow.c

```c
   printf("FOLLOW(%c) = { ",c);
   for(i=0;i<m;i++)
    printf("%c ",followResult[i]);
   printf(" }\n");
   printf("Do you want to continue(Press 1 to continue....)?");
  scanf("%d",&choice);
 }
 while(choice==1);
}
void follow(char c)
{
    if(a[0][0]==c)addToResult('$');
 for(i=0;i<n;i++)
 {
  for(j=2;j<strlen(a[i]);j++)
  {
   if(a[i][j]==c)
   {
    if(a[i][j+1]!='\0')first(a[i][j+1]);
    if(a[i][j+1]=='\0')//&&c!=a[i][0])
     follow(a[i][0]);
   }
  }
 }
}
void first(char c)
```

first.c folow.c

```
50  void first(char c)
51  {
52        int k;
53                  if(!(isupper(c)))
54                      //f[m++]=c;
55                      addToResult(c);
56                  for(k=0;k<n;k++)
57                  {
58                  if(a[k][0]==c)
59                  {
60                  if(a[k][2]=='$') follow(a[k][0]);
61                  else if(islower(a[k][2]))
62                      //f[m++]=a[k][2];
63                      addToResult(a[k][2]);
64                  else first(a[k][2]);
65                  }
66                  }
67  }
68  void  addToResult(char c)
69  {
70      int i;
71      for( i=0;i<=m;i++)
72          if(followResult[i]==c)
73              return;
74     followResult[m++]=c;
75  }
```

Output:

```
C:\Users\SHINY\Documents\lex_programs\folow.exe
Enter the no.of productions: 5
 Enter 5 productions
Production with multiple terms should be give as separate productions

S-aABb
A-c
A-$
B-d
B-$
Find FOLLOW of -->S
FOLLOW(S) = { $  }
Do you want to continue(Press 1 to continue....)?1
Find FOLLOW of -->A
FOLLOW(A) = { d b  }
Do you want to continue(Press 1 to continue....)?1
Find FOLLOW of -->B
FOLLOW(B) = { b  }
Do you want to continue(Press 1 to continue....)?3

--------------------------------
```

8. Calculator program using yacc tool

Program:

calc.l - Notepad

File Edit Format View Help

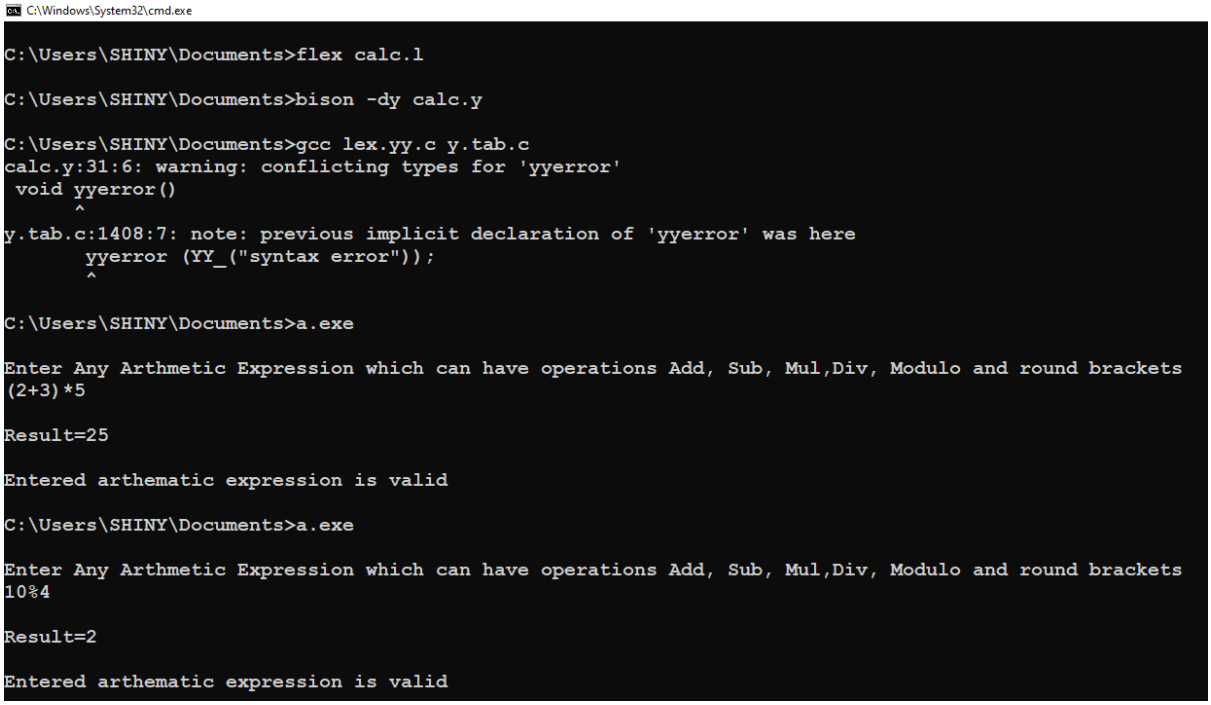
```
%{
#include<stdio.h>
#include "y.tab.h"
extern int yylval;
%}
%%
[0-9]+ {yylval=atoi(yytext); return NUMBER;}
[\t] ;
[\n] return 0;
. return yytext[0];
%%
int yywrap()
{
        return 1;
}
```

calc.y - Notepad

File Edit Format View Help

```
%{
#include<stdio.h>
int flag=0;
%}
%token NUMBER
%left '+' '-'
%left '*' '/' '%'
%left '(' ')'
%%
ArthmeticExpression: 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 Any Arthmetic Expression which can have operations Add, Sub, Mul,Div, Modulo and round brackets\n");
yyparse();
if(flag==0){
printf("\nEntered arthematic expression is valid\n");
}
}
void yyerror()
{
printf("\nEntered arthematic expression is Invalid\n\n");
flag=1;
}
```

Output:


C:\Windows\System32\cmd.exe
C:\Users\SHINY\Documents>flex calc.l
C:\Users\SHINY\Documents>bison -dy calc.y
C:\Users\SHINY\Documents>gcc lex.yy.c y.tab.c
calc.y:31:6: warning: conflicting types for 'yyerror'
void yyerror()
^
y.tab.c:1408:7: note: previous implicit declaration of 'yyerror' was here
yyerror (YY_("syntax error"));
^
C:\Users\SHINY\Documents>a.exe
Enter Any Arthmetic Expression which can have operations Add, Sub, Mul,Div, Modulo and round brackets
(2+3)*5
Result=25
Entered arthematic expression is valid
C:\Users\SHINY\Documents>a.exe
Enter Any Arthmetic Expression which can have operations Add, Sub, Mul,Div, Modulo and round brackets
10%4
Result=2
Entered arthematic expression is valid