# Compiler Design Lab

## Programs list

1. Write a LEX program to find number of occurrences of given string in a file.
2. Write a LEX program to insert line numbers in a file.
3. Write a LEX program to find the number of lines, characters, words in the input file.
4. Write a LEX program to find the successor, predecessor of the input.
5. Write a LEX program to print substrings of length 2 of a given string.
6. Write a LEX program to find reverse of a given string.
7. Write a LEX program to remove the comments in the given file.
8. Write a LEX program to find the frequency of vowels in a file.
9. Write a LEX program which extracts integer type of numbers, float type of numbers, keywords and identifiers in a program.
10. Write a program to find the FIRST of a Non Terminal in a given grammar.(you may need to take non-terminals and terminals by standard input )
11. Write a C program to check whether the given string is accepted by the given grammar by using Recursive Descent Parsing.
12. Write a YACC program to convert a decimal number to binary.
13. Write a YACC program to convert a binary number to decimal.
14. Write a YACC program for balanced parenthesis.
15. Write a YACC program for validity of arithmetic expressions.
16. Write a YACC program for anbn grammar.
17. Write a YACC program for date validation.

**Write a LEX program to print substrings of length 2 of a given string.**

%{

int i;

%}

%%

.\* {for(i=0;i<yyleng-1;i++)

{

printf("%c%c\t",yytext[i],yytext[i+1]);

}

}

%%

int main(int argc, char \*argv[])

{

extern FILE \*yyin;

yyin=fopen(argv[1],"r");

yylex();

return 0;

}

int yywrap(){

return 1;

}

**Write a LEX program to find the successor, predecessor of the input.**

%{

char i;

%}

%%

[b-yB-Y] { i=yytext[0];

printf("predecessor:%c\t successor:%c",i-1,i+1);

}

[0-9]+ {

printf("predecessor:%d\t",atoi(yytext)-1);

printf("successor:%d\t",atoi(yytext)+1); }

[a|A] {i=yytext[0];printf("no predecessor \t successor:%c",i+1);}

[z|Z] {i=yytext[0];printf("no successor \t predecessor:%c",i-1);}

%%

main(int argc ,char \*argv[])

{

extern FILE \*yyin;

yyin=fopen(argv[1],"r");

yylex();

return 0;

}

int yywrap()

{

return 1;

}

**Write a LEX program to find reverse of a given string.**

%{

int i;

%}

%%

.\* {

for(i=yyleng;i>=0;i--)

{

printf("%c",yytext[i]);

}

}

%%

int main(){

printf("Enter some text here:");

yylex();

}

int yywrap()

{

return 1;

}

**Write a YACC program to convert a decimal number to binary.**

**.lex file**

%{

#include<stdio.h>

#include "y.tab.h"

extern int yylval;

%}

%%

[0-9]+ {yylval=atoi(yytext);

return num;}

%%

**.y file**

%{

#include<stdio.h>

#include<math.h>

int i=0,r=0,a[10],x,j=0;

%}

%token num;

%%

start:num{

x=$1;

while(x>0){

r=x%2;

a[i]=r;

x=x/2;

i++;}

for(j=i-1;j>=0;j--){

printf("%d\t",a[j]);

}

}

;

%%

main( int argc, char \*argv[] )

{ extern FILE \*yyin;

yyin = fopen( argv[1], "r" );

yyparse ();

}

yyerror ( char \*s )

{

printf("");

}

yywrap()

{

return(1);

}

**Write a YACC program to convert a binary number to decimal.**

**.lex file**

%{

#include<stdio.h>

#include "y.tab.h"

extern int yylval;

%}

%%

[01]+ {yylval = atoi(yytext); return NUM;}

[0-9]+ {printf("\n\tEnter binary digits only !!!!!!?\n\n");exit(0);}

\n {return NL;}

%%

int yywrap()

{

return 1;

}

**.y file**

%{

#include<stdio.h>

void yyerror(char \*s);

%}

%token NUM

%token NL

%token SYM

%%

E : NUM NL {binary($1);exit(0);}

;

%%

void yyerror(char \*s)

{

printf("\nFormat error, Enter the binary digits only\n");

getchar();

}

int main()

{

yyparse();

getchar();

}

int binary(int n)

{

int sum=0,a=0,temp=1;

while(n>0)

{

a=n%10;

sum=sum+(temp\*a);

temp=temp\*2;

n=n/10;

}

printf("\n\tThe decimal format of the given binary Number:- %d\n\n",sum);

}

**Write a program to find the FIRST of a Non Terminal in a given grammar.(you may need to take non-terminals and terminals by standard input )**

#include<stdio.h>

#include<string.h>

main()

{

int i,j,nt,t,p;

printf("Enter the no of terminals :");

scanf("%d",&t);

printf("\n");

char te[t];

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

{

printf("Enter the %d terminal:",i+1);

scanf("%s",&te[i]);

}

printf("Enter the no of non terminals:");

scanf("%d",&nt);

char nte[nt];

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

{

printf("Enter the %d non terminal:",i+1);

scanf("%s",&nte[i]);

}

printf("Enter the no of productions you have:");

scanf("%d",&p);

printf("Enter # at end of the production:\n");

char pr[p][10],fir[0],ch,rch;

int nti,c=0;

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

{

for(j=0;j<10;j++)

{ if(j==0)

{

printf("Enter the left hand side production:");

}

else

{ printf("Enter the right hand side production one symbol at a time:");

}

scanf("%s",&pr[i][j]);

if( pr[i][j]=='#')

break;

}

}

goto AR;

AR :

printf("Enter the non terminal for which you want to find the first:");

scanf("%s",&ch);

rch=ch;

goto AE;

AE :

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

{

if(ch==pr[i][0])

nti = i;

else

i++;

for(j=0;j<10;j++)

if(pr[nti][1]==te[j])

{ fir[0] = te[j];

}

else

if(pr[nti][1]== nte[j])

{

ch = nte[j];

}

}

c++;

printf("the first of %c is %c\n",rch,fir[0]);

if(c<nt)

{ goto AR;

}

else

{ printf("\n....The program is over.......\n");

}

}