1.Problem Statement : Lex programs to recognize Keywords

Source Code:

%{

#include<stdio.h>

%}

%%

int|float|char { printf(" data type: %s",yytext); }

%%

main()

{

yylex();

return(0);

}

Execution:

Input:int

Output:datatype:int

2.Problem Statement : Lex programs to recognize String ending with 00.

Source Code:

%%

[0-9]\*00{printf(“string accepted”);

[0-9]\*{printf(“string rejected”);}

%%

main()

{

yylex();

}

int yywrap()

{

return 1;

}

Execution:

Input: **10100**

Output: **String accepted**

3.Problem Statement : Program to recognize the strings which are starting or ending with ‘k’.

Source Code:

%{

#include<stdio.h>

%}

%%

k[a-z]\*[A-Z]\*[0-9]\* | [a-z]\*[A-Z]\*[0-9]\*k { printf("matching"); }

.\* { printf("not matching"); }

%%

main()

{

yylex();

return 0;

}

int yywrap()

{

}

Execution:

Input: **kanna**

Output: matching

4.Problem Statement : program to assign line numbers for source code.

Source Code:

%{

#include&

#include&

int ln=0;

%}

%%

"\n" {}

.\* {ln++;fprintf(yyout,"\n%d:%s",ln,yytext);}

%%

main()

{

yyin=fopen("try1.txt","r");

yyout=fopen("try2.txt","w");

yylex();

return 0;

}

int yywrap()

{

}

Execution:

try1.txt

hai  
hello world  
test file

language processor

try2.txt

1:hai  
2:hello world

3:test file  
4:language processor

5.Problem Statement : Program to recognize the numbers which has 1 in its 5th position from right.

Source Code:

%{

#include<stdio.h>

%}

%%

.\*1….{ printf("matching"); }

.\* { printf("not matching"); }

%%

main()

{

yylex();

return 0;

}

int yywrap()

{

}

Execution:

Input: **210303**

Output: matching

6.Problem Statement : Implement lexical analyzer in Lex.

%{  
int COMMENT=0;  
%}  
identifier [a-zA-Z][a-zA-Z0-9]\*  
%%  
#.\* {printf("\n%s is a preprocessor directive",yytext);}  
int |  
float |  
char |  
double |  
while |  
for |  
struct |  
typedef |  
do |  
if |  
break |  
continue |  
void |  
switch |  
return |  
else |  
goto {printf("\n\t%s is a keyword",yytext);}  
"/\*" {COMMENT=1;}{printf("\n\t %s is a COMMENT",yytext);}  
{identifier}\( {if(!COMMENT)printf("\nFUNCTION \n\t%s",yytext);}  
\{  {if(!COMMENT)printf("\n BLOCK BEGINS");}  
\}  {if(!COMMENT)printf("BLOCK ENDS ");}  
{identifier}(\[[0-9]\*\])? {if(!COMMENT) printf("\n %s IDENTIFIER",yytext);}  
\".\*\" {if(!COMMENT)printf("\n\t %s is a STRING",yytext);}  
[0-9]+ {if(!COMMENT) printf("\n %s is a NUMBER ",yytext);}  
\)(\:)? {if(!COMMENT)printf("\n\t");ECHO;printf("\n");}  
\( ECHO;  
= {if(!COMMENT)printf("\n\t %s is an ASSIGNMENT OPERATOR",yytext);}  
\<= |  
\>= |  
\< |  
== |  
\> {if(!COMMENT) printf("\n\t%s is a RELATIONAL OPERATOR",yytext);}  
%%

int main(int argc, char \*\*argv)  
{  
FILE \*file;  
file=fopen("var.c","r");  
if(!file)  
{  
printf("could not open the file");  
exit(0);  
}  
yyin=file;  
yylex();  
printf("\n");  
return(0);  
}  
int yywrap()  
{  
return(1);  
}

**INPUT:**  
//var.c  
#include<stdio.h>  
#include<conio.h>  
void main()  
{  
int a,b,c;  
a=1;  
b=2;  
c=a+b;  
printf("Sum:%d",c);  
}

**OUTPUT:**

[Text

Description automatically generated](https://4.bp.blogspot.com/-hv4TlVO-pI8/WiZDkINhW-I/AAAAAAAAC3o/q5aM-MuAvCIO-tdoc3BqmS4Toa-8UJ8pgCLcBGAs/s1600/exp3.PNG)