VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)
(Affiliated to Osmania University)
Hyderabad-500031

DEPARTMENT OF : _____

NAME OF THE LABORATORY: _____

Name:_____ HT No.:_____ Page No.:_____

WEEK-2

1.) Lex programs to recognize keywords.

```
%{
#include<stdio.h>
%}
%%
int|float|char { printf("data type:%s", yytext);}
%%
int main()
{ yylex(); return(0);}
int yywrap() { }.
```

OUTPUT:

```
int
data type:int
```

2.) Lex programs to recognize string ending with 00.

```
%{
#include<stdio.h> %}
%%
[a-z A-Z 0-9]+00 { printf("string is accepted", yytext);}
.* { printf("not accepted", yytext);}
%%
int main() {
```

```
yylex();
return(0); }
int yywrap(){ }
```

OUTPUT:

```
prabarshitha00
String is accepted
```

(3.) Program to recognize the strings which are starting or ending with 'k'

```
%{
#include<stdio.h>
%}
begin-with-k k.*
end-with-k.*k
%%
{begin-with-k} {printf("%s is a word that begin with k", yytext);}
{end-with-k} {printf("%s is a word that end with k", yytext);} %%
int main(){ yylex(); return(0);
}
int yywrap()
{ }
```

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)
(Affiliated to Osmania University)
Hyderabad-500031

DEPARTMENT OF : _____

NAME OF THE LABORATORY: _____

Name:_____ HT No.:_____ Page No.:_____

OUTPUT:

```
kcompile
k compile is a word that begin with k.
```

---

4) Program to assign line numbers for source code.

```
%{
#include <stdio.h>
int lineno=0; %}
line.* \n
%%
{line} { printf("%d.%s",lineno++,yytext);}%%
int main(int argc, char **argv)
{
if (argc>1)
{
FILE *file;
file = fopen (argv[1],"r");
if (!file)
{ printf("could not open %s\n",argv[1]);
exit(0); }
yyin=file; }
```

```
yylex();
printf("\n\n");
return 0; }
int yywrap()
{ return 0;}
```

file
abcd
efgh
ijkl

OUTPUT:
1 abcd
2 efgh
3 ijkl

(5) Program to recognize the numbers which has 1 in its 5th position from right.

```
%{
#include<stdio.h>%}
%%
[0-9]*1[0-9]{4}  { printf("accepted:"); }
[0-9]*1[0-9]     { printf("NOT accepted:"); }
%%
int main()
{ yylex();
  return(0); }
int yywrap()
{ }.
```

OUTPUT:

00010000

accepted.

---

6. Write lex specification to replace space with $

```
%{
#include <stdio.h> %}.
%%
[ ] {printf("$ ");}
.|\n {printf("%s", yytext);}
%%
int main() {
        yylex();
        return 0; }
int yywrap()
{
}.
```

OUTPUT:

abc def

abc$def

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)
(Affiliated to Osmania University)
Hyderabad-500031

DEPARTMENT OF : _____

NAME OF THE LABORATORY: _____

Name:_____ HT No.:_____ Page No.:_____

7. Write lex specification to identify number divide by 9

```
%{
#include <stdio.h>
%}
%%
^[1-9][0-9]*[0,9]$ {if(atoi(yytext)%9==0)printf("%s is
divisible by 9 \n",yytext);}
0                  {printf("0 is divisible by 9");}
%%
int yywrap(){  return 1;}
int main(){ yylex(); return 0;}
```

OUTPUT:

```
999
999 is divisible by 9
```

---

8. Write lex specification identify identifier and function -al identifier.

```
%{ #include <stdio.h> %}
DIGIT    [0-9]
LETTER   [a-zA-Z]
ID       {LETTER}{LETTER}|{DIGIT})*
FUNC-ID  {ID}\([^)]*\)
```

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)
(Affiliated to Osmania University)
Hyderabad-500031

DEPARTMENT OF : _____

NAME OF THE LABORATORY: _____

Name:_____ HT No.:_____ Page No.:_____

```
%%
{FUNC_ID} {printf("Functional Identifier: %s\n",yytext);
 {ID}     {printf("Identifier: %s\n", yytext);}
%%
int yywrap() { return 1; }
int main() { yylex(); return 0; }
```

OUTPUT:

```
xyz2012
Identifier: xyz2012
```

(9.) write lex specification identify comment..

```
%{
#include<stdio.h> %}
%%
"//".*\n  {printf("Single-line comment: %s\n",yytext);}
"/*"([^*]|\*+[^*/])*"*"+"/"{printf("Multi-Line coment: %s\n
                                ",yytext);}
%%
int yywrap(){ return 1; }
int main(){
    yylex();
    return 0;
}
```

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)
(Affiliated to Osmania University)
Hyderabad-500031

DEPARTMENT OF : _____

NAME OF THE LABORATORY: _____

Name:_____ HT No.:_____ Page No.:_____

OUTPUT:

// xyz //

Single - Line comment : //

(10.) Write lex specification to identify real number.

```
%{ #include<stdio.h> %}
DIGIT [0-9]
EXPONENT [eE][+-]?{DIGIT}+
REAL_NUMBER {DIGIT}+(\.{DIGIT}+)?({EXPONENT})?
%%
{REAL_NUMBER}{printf("Real number: %s\n", yytext);}
%%
int yywrap(){ return 1; }
int main(){ yylex() return 0; }.
```

OUTPUT:

```
0.123
Real number: 0.123
```