LP LAB ASSIGNMENT – 3

SUBMITTED BY: TEJASWO TIWARY
ROLL NO. 207278
SECTION: C

1. LEX program to recognises the keyword if, begin and identifier which is defined as any string starts with letter and followed by letter or digit.

CODE:

q1.l

home > tejaswo > Downloads > lp > q1.l

```
%{
#include<bits/stdc++.h>
using namespace std;
%}
%%
    //"\n" {return 0;}
"if"      {cout<<"if is keyword \n";}
begin   {cout<<"begin is keyword\n";}
[a-zA-Z]+[a-zA-Z0-9]* {cout<<"identifier\n";}
.*  {cout<<"others\n";}
%%
int yywrap() {
    return 1;
}
int main() {
    yylex();
    return 0;
}

```

OUTPUT:

tejaswo@tejaswo: ~/Downloads/lp

```
tejaswo@tejaswo:~/Downloads/lp$ lex q1.l
tejaswo@tejaswo:~/Downloads/lp$ g++ lex.yy.c -ll
tejaswo@tejaswo:~/Downloads/lp$ ./a.out
if
if is keyword

begin
begin is keyword

abc
identifier

1
others

^C
tejaswo@tejaswo:~/Downloads/lp$
```

2. LEX program to recognises the keyword if, begin and identifier which is defined as any string starts with letter and followed by letter or digit and count the number of identifiers, keywords and operators encountered in the input.

CODE:

q2.l

home > tejaswo > Downloads > lp > q2.l

```
%{
#include<bits/stdc++.h>
using namespace std;
int k=0,i=0,op=0;
%}
%%
    //"\n" {return 0;}
"if"      {k++;}
begin   {k++;}
[a-zA-Z]+[a-zA-Z0-9]* {i++;}
\+   {op++;}
\/   {op++;}
\*   {op++;}
\-   {op++;}
\%   {op++;}
.*  {return 0;}
%%
int yywrap() {
    return 1;
}
int main() {
    yylex();
    cout<<"No. of keywords: "<<k<<"\n";
    cout<<"No. of identifier: "<<i<<"\n";
    cout<<"No. of operators: "<<op<<"\n";
    return 0;
}
```

OUTPUT:

tejaswo@tejaswo: ~/Downloads/lp

```
tejaswo@tejaswo:~/Downloads/lp$ lex q2.l
tejaswo@tejaswo:~/Downloads/lp$ g++ lex.yy.c -ll
tejaswo@tejaswo:~/Downloads/lp$ ./a.out
if

begin

a12

aa

if

12
No. of keywords: 3
No. of identifier: 2
No. of operators: 0
tejaswo@tejaswo:~/Downloads/lp$
```

3. Lex program to recognise whether a given sentence is simple or compound.

CODE:

q3.l

home > tejaswo > Downloads > lp > q3.l

```
%{
    #include<stdio.h>
    int flag=0;
%}

%%
and |
or |
but |
because |
if |
then |
nevertheless  { flag=1; }
.  ;
\n  { return 0; }
%%

int main()
{
    printf("Enter the sentence:\n");
    yylex();
    if(flag==0)
        printf("Simple sentence\n");
    else
        printf("compound sentence\n");
}

int yywrap( )
{
    return 1;
}
```

OUTPUT:

tejaswo@tejaswo: ~/Downloads/lp

```
tejaswo@tejaswo:~/Downloads/lp$ lex q3.l
tejaswo@tejaswo:~/Downloads/lp$ g++ lex.yy.c -ll
tejaswo@tejaswo:~/Downloads/lp$ ./a.out
Enter the sentence:
my name is tejaswo
Simple sentence
tejaswo@tejaswo:~/Downloads/lp$ ./a.out
Enter the sentence:
my name is tejaswo and i am studying CSE
compound sentence
tejaswo@tejaswo:~/Downloads/lp$
```

4. Lex program to count the frequency of the given word in a file.

CODE:

q4.l

home > tejaswo > Downloads > lp > q4.l

```
%{
#include <stdio.h>
#include <string.h>
char word[]="the";
int count=0;
%}
%%
[a-zA-Z]+ {if (strcmp(word,yytext)==0){count++;}}
. ;
%%
int yywrap(){ return 1;}
int main (){
extern FILE *yyin, *yyout;
yyin=fopen ("input.txt","r");
yylex();
printf("count of the word '%s' in file is: %d\n",word,count);
}
```

OUTPUT:

tejaswo@tejaswo: ~/Downloads/lp

```
tejaswo@tejaswo:~/Downloads/lp$ lex q4.l
tejaswo@tejaswo:~/Downloads/lp$ g++ lex.yy.c -ll
tejaswo@tejaswo:~/Downloads/lp$ ./a.out
count of the word 'the' in file is: 12
tejaswo@tejaswo:~/Downloads/lp$
```

5. Lex program to check perfect numbers. Perfect number, a positive integer that is equal to the sum of its proper divisors, for example: 6 = 1+2+3.

CODE:

q5.l

home > tejaswo > Downloads > lp > q5.l

```
%{
#include<string.h>
void check(char *);
%}
%%
[0-9]+ check(yytext);
%%
int main()
{

    yylex();
}
void check(char *a)
{
    int len=strlen(a),i,num=0;
    for(i=0;i<len;i++)
        num=num*10+(a[i]-'0');
    int x=0,temp=num;
    for(i=1;i<num;i++)
    {
        if(num%i==0)
            x=x+i;
    }
    if(x==temp)
        printf("%d is perfect \n",num);
    else
        printf("%d is not perfect \n",num);
}
```

OUTPUT:

tejaswo@tejaswo: ~/Downloads/lp

```
tejaswo@tejaswo:~/Downloads/lp$ lex q5.l
tejaswo@tejaswo:~/Downloads/lp$ g++ lex.yy.c -ll
tejaswo@tejaswo:~/Downloads/lp$ ./a.out
128
128 is not perfect

6
6 is perfect

8128
8128 is perfect

^C
tejaswo@tejaswo:~/Downloads/lp$
```

6. Write LEX Code that accepts the string having even number's of 'a' over input alphabet {a, b}.

CODE:

q6.l

home > tejaswo > Downloads > lp > q6.l

```
%{
%}

%s A DEAD

%%
<INITIAL>a BEGIN A;
<INITIAL>b BEGIN INITIAL;
<INITIAL>[^ab\n] BEGIN DEAD;
<INITIAL>\n BEGIN INITIAL; {printf("Accepted\n");}

<A>a BEGIN INITIAL;
<A>b BEGIN A;
<A>[^ab\n] BEGIN DEAD;
<A>\n BEGIN INITIAL; {printf("Not Accepted\n");}

<DEAD>[^\n] BEGIN DEAD;
<DEAD>\n BEGIN INITIAL; {printf("Invalid\n");}

%%

int yywrap()
{
return 1;
}
int main()
{
   printf("Enter String\n");
   yylex();
return 0;
}
```

OUTPUT:

tejaswo@tejaswo: ~/Downloads/lp

```
tejaswo@tejaswo:~/Downloads/lp$ lex q6.l
tejaswo@tejaswo:~/Downloads/lp$ g++ lex.yy.c -ll
tejaswo@tejaswo:~/Downloads/lp$ ./a.out
Enter String
aaaa
Accepted
a
Not Accepted
aaaabbb
Accepted
abababa
Accepted
aaaaa
Not Accepted
^C
tejaswo@tejaswo:~/Downloads/lp$
```

7. Write LEX code which accepts Odd number of 0’s and even number of 1’s.

CODE:

q7.l

home > tejaswo > Downloads > lp > q7.l

```
%{
%}

%s A B C DEAD

%%
<INITIAL>1 BEGIN A;
<INITIAL>0 BEGIN B;
<INITIAL>[^01\n] BEGIN DEAD;
<INITIAL>\n BEGIN INITIAL; {printf("Not Accepted\n");}

<A>1 BEGIN INITIAL;
<A>0 BEGIN C;
<A>[^01\n] BEGIN DEAD;
<A>\n BEGIN INITIAL; {printf("Not Accepted\n");}

<B>1 BEGIN C;
<B>0 BEGIN INITIAL;
<B>[^01\n] BEGIN DEAD;
<B>\n BEGIN INITIAL; {printf("Accepted\n");}

<C>1 BEGIN B;
<C>0 BEGIN A;
<C>[^01\n] BEGIN DEAD;
<C>\n BEGIN INITIAL; {printf("Not Accepted\n");}

<DEAD>[^\n] BEGIN DEAD;
<DEAD>\n BEGIN INITIAL; {printf("Invalid\n");}

%%

int main()
{
printf("Enter String\n");
yylex();return 0;
}
```

OUTPUT:

```
tejaswo@tejaswo: ~/Downloads/lp
tejaswo@tejaswo:~/Downloads/lp$ lex q7.l
tejaswo@tejaswo:~/Downloads/lp$ g++ lex.yy.c -ll
tejaswo@tejaswo:~/Downloads/lp$ ./a.out
Enter String
00011
Accepted
00111
Not Accepted
000001111
Accepted
^C
tejaswo@tejaswo:~/Downloads/lp$
```

8. Write LEX code which accepts strings ending with 11.

CODE:

q8.l

home > tejaswo > Downloads > lp > q8.l

```
%{
%}

%s A B DEAD
%%
<INITIAL>1 BEGIN A;
<INITIAL>0 BEGIN INITIAL;
<INITIAL>[^01\n] BEGIN DEAD;
<INITIAL>\n BEGIN INITIAL; {printf("Not Accepted\n");}

<A>1 BEGIN B;
<A>0 BEGIN INITIAL;
<A>[^01\n] BEGIN DEAD;
<A>\n BEGIN INITIAL; {printf("Not Accepted\n");}

<B>1 BEGIN B;
<B>0 BEGIN INITIAL;
<B>[^01\n] BEGIN DEAD;
<B>\n BEGIN INITIAL; {printf("Accepted\n");}
<DEAD>[^\n] BEGIN DEAD;
<DEAD>\n BEGIN INITIAL; {printf("Invalid\n");}
%%

int main()
{
printf("Enter String\n");
yylex();
return 0;
}
```

OUTPUT:

```
tejaswo@tejaswo: ~/Downloads/lp
tejaswo@tejaswo:~/Downloads/lp$ lex q8.l
tejaswo@tejaswo:~/Downloads/lp$ g++ lex.yy.c -ll
tejaswo@tejaswo:~/Downloads/lp$ ./a.out
Enter String
000011
Accepted
00001
Not Accepted
00000111
Accepted
^C
tejaswo@tejaswo:~/Downloads/lp$
```

9. Lex Program For checking a valid URL.

CODE:

q9.l

home > tejaswo > Downloads > lp > q9.l

```
%{
#include<stdio.h>
#include<stdlib.h>
int flag=0;
%}
%%
[a-z . 0-9]+[a-z]+".com"|".in" { flag=1; }
%%
int main()
{
yylex();
if(flag==1)
printf("Accepted\n");
else
printf("Not Accepted\n");
}
```

OUTPUT:

tejaswo@tejaswo: ~/Downloads/lp

```
tejaswo@tejaswo:~/Downloads/lp$ lex q9.l
tejaswo@tejaswo:~/Downloads/lp$ g++ lex.yy.c -ll
tejaswo@tejaswo:~/Downloads/lp$ ./a.out
abcd.com

Accepted
tejaswo@tejaswo:~/Downloads/lp$ ./a.out
abcd
abcd
abcd.nitw.ac.in
abcd.nitw.ac
Accepted
tejaswo@tejaswo:~/Downloads/lp$ ./a.out
abcd
abcd
Not Accepted
tejaswo@tejaswo:~/Downloads/lp$
```

10. Lex Program to check valid email.

CODE:

q10.l

home > tejaswo > Downloads > lp > q10.l

```
%{
#include<stdio.h>
#include<stdlib.h>
int flag=0;
%}
%%
[a-z . 0-9]+@[a-z]+".com"|".in" { flag=1; }
%%
int main()
{
yylex();
if(flag==1)
printf("Accepted\n");
else
printf("Not Accepted\n");
}
```

OUTPUT:

tejaswo@tejaswo: ~/Downloads/lp

```
tejaswo@tejaswo:~/Downloads/lp$ lex q10.l
tejaswo@tejaswo:~/Downloads/lp$ g++ lex.yy.c -ll
tejaswo@tejaswo:~/Downloads/lp$ ./a.out
abcd@gmail.com

Accepted
tejaswo@tejaswo:~/Downloads/lp$ ./a.out
abcd
abcd
Not Accepted
tejaswo@tejaswo:~/Downloads/lp$
```