CSB 353: Compiler Design

LAB 4

Submitted By:

Name: PREM KUMAR

Roll No: 191210037

Branch: CSE

Semester: 6 th

Submitted To: Dr. Shelly Sachdeva

Department of Computer Science and Engineering

NATIONAL INSTITUTE OF TECHNOLOGY DELHI

2019-2023

Ques 1. Consider the simplest regular expressions which are the strings of text characters with no operators at all like monday tuesday wednesday. These three regular expressions match any occurrences of those character strings in an input text. Design a Scanner that removes every occurrence of the word day in such words.

Code:

```
%{
#include<stdio.h>
%}
%%
"day" ;
%%

main(void)
{
yyin= fopen("input.txt","r");
yyout= fopen("output.txt","w");
yylex();

}

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

Input:

```
monday tuesday wednesday
thursday
friday saturday
sunday
```

Output:

```
mon tues wednes
thurs
fri satur
sun
```

Ques 2. Design a Lexical analyzer which is successfully able to execute the following tasks:

(i) Count the total number of all digit strings in an input text

(ii) Print the running total of the number of digit strings

(iii) Print out each one as soon as it is found.

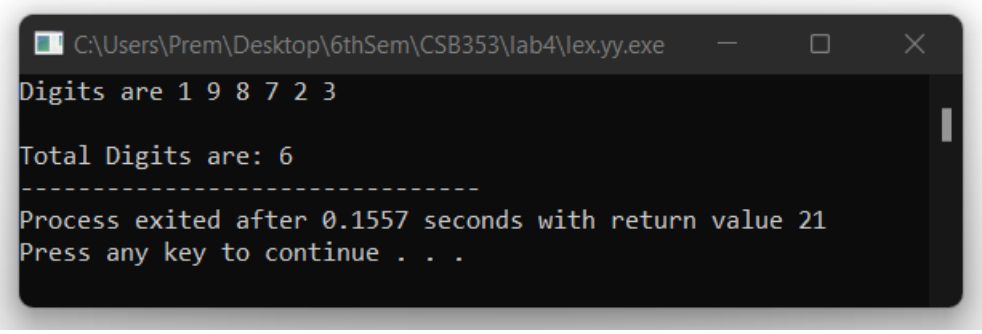
Code:

```
%{
#include<stdio.h>
int digits=0,val=0;
%}
DIGIT    [0-9]
%%
{DIGIT} {
    if(digits==0)
    printf("Digits are ");
    printf("%s ", yytext);
    digits++;
}
. {
    printf("");
}
%%

main(void)
{
yyin= fopen("input2.txt","r");

yylex();

printf("\n\nTotal Digits are: %d",digits);


}

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

Input:

```
1 9 8 7 2 3 abc
```

Output:



Ques 3. Design a Lexical analyzer which is successfully able to execute the following tasks:

(i) Count the total number of all digit strings in an input text

(ii) Print the running total of the number of digit strings

(iii) Print out each one as soon as it is found.

Code:

```
%{
#include<stdio.h>
%}
%%
"MOVE"|"LEFT"|"RIGHT"|"UP"|"DOWN" {
    printf("\nToken: %s ", yytext);
}
[0-9]|[1-9][0-9]* {
    printf("\nToken: %s ", yytext);
}
[ \t\n]+
. {
    printf("\nInvalid token: %s",yytext);
}
%%

main(void)
{
yyin= fopen("input3.txt","r");

yylex();



}

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

Input:

```
MOVE LEFT 3
MOVE UP 1
MOVE RIGHT 10
MOVE LEFT 5
MOVE DOWN 4
```

Output: