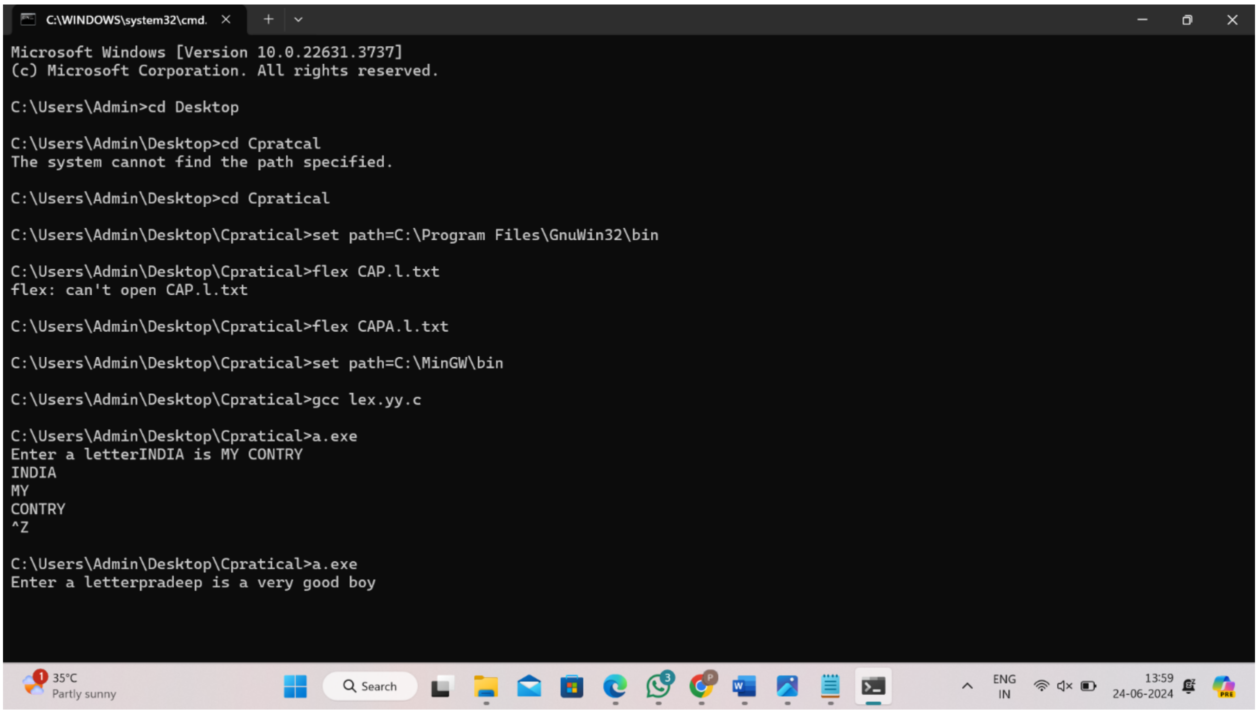
DAY-1 Experiments

1.Write a LEX program to identify the capital words from the given input.

Program:

```
%{
%}
%%
[A-Z] {printf("%c",yytext[0]);}
. {}
%%
int yywrap(){}
int main()
{
printf("\nenter the string:");
yylex();
}
```

Output:



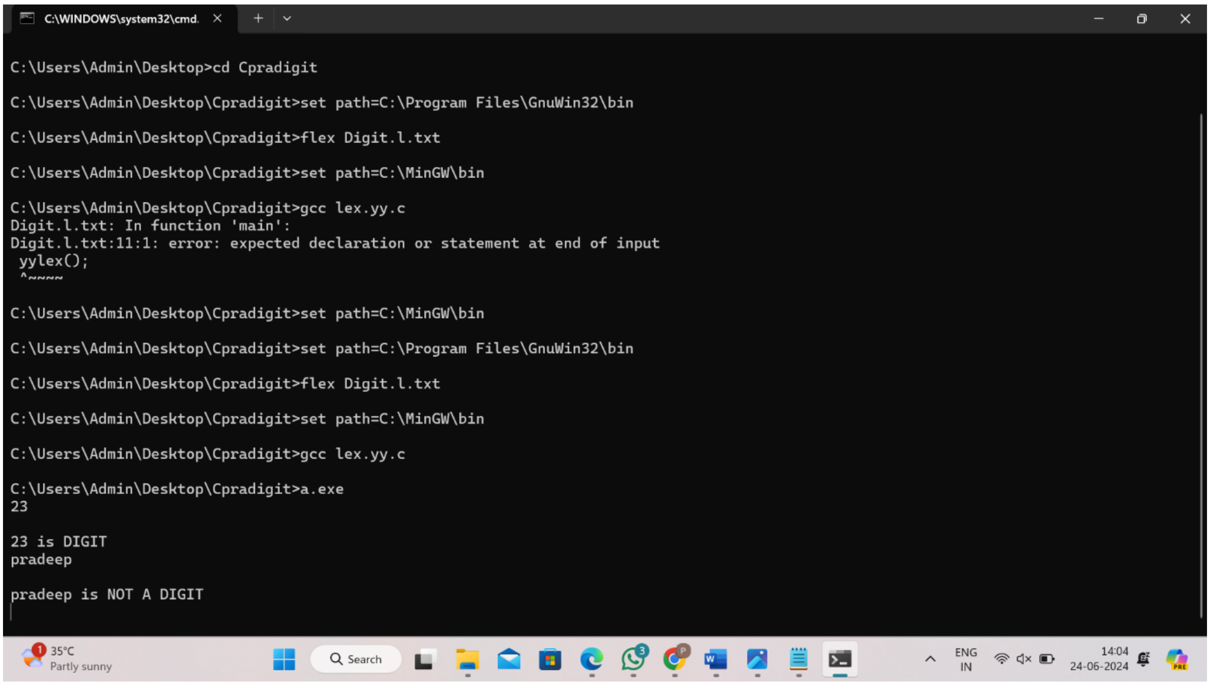
```
Microsoft Windows [Version 10.0.22631.3737]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin>cd Desktop

C:\Users\Admin\Desktop>cd Cpratcal
The system cannot find the path specified.

C:\Users\Admin\Desktop>cd Cpratical

C:\Users\Admin\Desktop\Cpratical>set path=C:\Program Files\GnuWin32\bin

C:\Users\Admin\Desktop\Cpratical>flex CAP.l.txt
flex: can't open CAP.l.txt

C:\Users\Admin\Desktop\Cpratical>flex CAPA.l.txt

C:\Users\Admin\Desktop\Cpratical>set path=C:\MinGW\bin

C:\Users\Admin\Desktop\Cpratical>gcc lex.yy.c

C:\Users\Admin\Desktop\Cpratical>a.exe
Enter a letterINDIA is MY CONTRY
INDIA
MY
CONTRY
^Z

C:\Users\Admin\Desktop\Cpratical>a.exe
Enter a letterpradeep is a very good boy
```

2. Write a LEX program to check whether the given input is digit or not.

Program:

```
%{
#include<stdio.h>
%}
%%
[0-9]+|[0-9]*\.[0-9]+ { printf("\n%s is DIGIT", yytext);}
.+ { printf("\n%s is NOT A DIGIT",yytext);}
%%
int yywrap(){}
int main()
{
yylex();
}
```

Output:



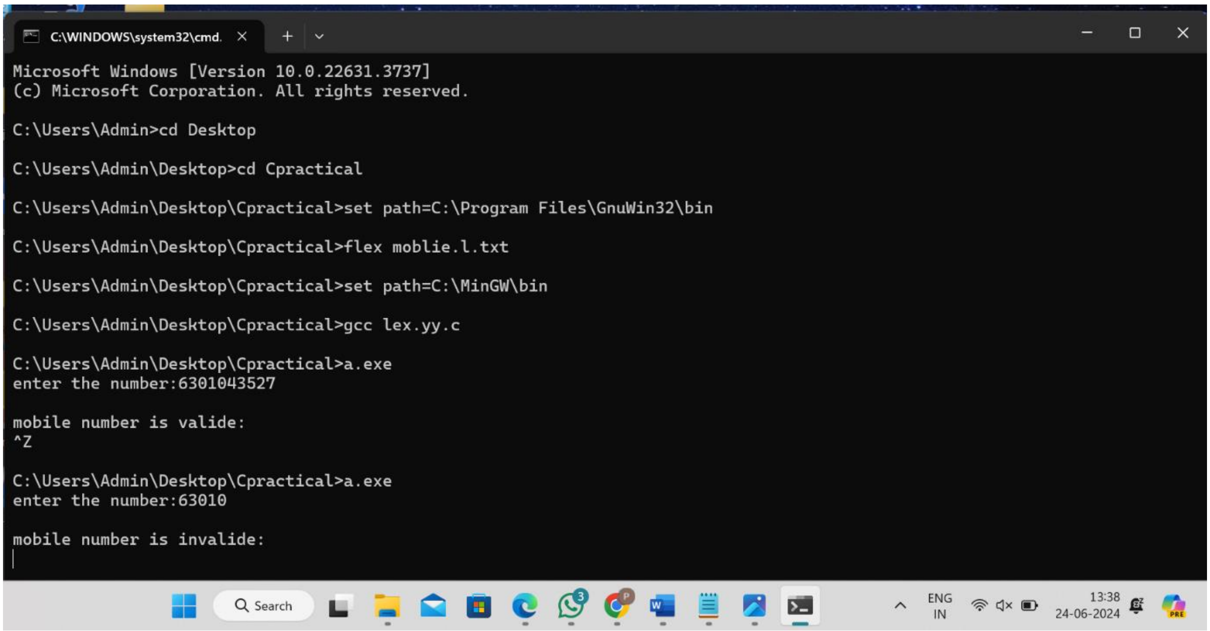
```
C:\Users\Admin\Desktop>cd Cpradigit

C:\Users\Admin\Desktop\Cpradigit>set path=C:\Program Files\GnuWin32\bin

C:\Users\Admin\Desktop\Cpradigit>flex Digit.l.txt

C:\Users\Admin\Desktop\Cpradigit>set path=C:\MinGW\bin

C:\Users\Admin\Desktop\Cpradigit>gcc lex.yy.c
Digit.l.txt: In function 'main':
Digit.l.txt:11:1: error: expected declaration or statement at end of input
 yylex();
 ^~~~~

C:\Users\Admin\Desktop\Cpradigit>set path=C:\MinGW\bin

C:\Users\Admin\Desktop\Cpradigit>set path=C:\Program Files\GnuWin32\bin

C:\Users\Admin\Desktop\Cpradigit>flex Digit.l.txt

C:\Users\Admin\Desktop\Cpradigit>set path=C:\MinGW\bin

C:\Users\Admin\Desktop\Cpradigit>gcc lex.yy.c

C:\Users\Admin\Desktop\Cpradigit>a.exe
23

23 is DIGIT
pradeep

pradeep is NOT A DIGIT
```

3. Implement a LEX program to check whether the mobile number is valid or not.

Program:

```
%{
%}
%%
[0-9][0-9]{9} {printf("\n mobile number valid\n");}
.+ {printf("\n mobile number invalid\n");}
%%
int yywap()
{}
int main()
{
printf("\n enter the mobile number:");
yylex();
}
```

Output:



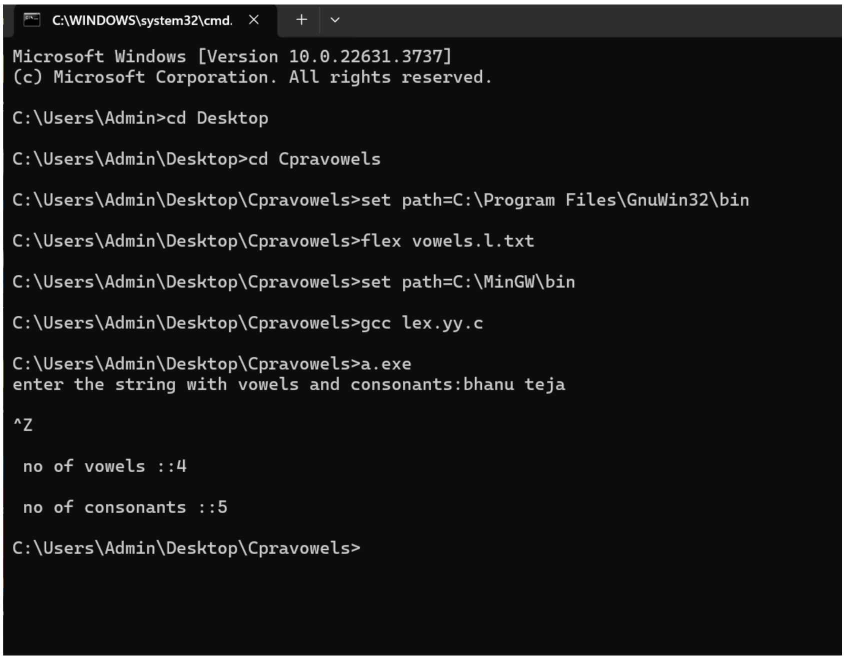
```
Microsoft Windows [Version 10.0.22631.3737]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin>cd Desktop

C:\Users\Admin\Desktop>cd Cpractical

C:\Users\Admin\Desktop\Cpractical>set path=C:\Program Files\GnuWin32\bin

C:\Users\Admin\Desktop\Cpractical>flex moblie.l.txt

C:\Users\Admin\Desktop\Cpractical>set path=C:\MinGW\bin

C:\Users\Admin\Desktop\Cpractical>gcc lex.yy.c

C:\Users\Admin\Desktop\Cpractical>a.exe
enter the number:6301043527

mobile number is valide:
^Z

C:\Users\Admin\Desktop\Cpractical>a.exe
enter the number:63010

mobile number is invalide:
```

4. Write an algorithm to help the student to count the number of vowels and consonants in the given sentence.

Program:

```
%{
int vcount=0;
int ccount=0;
%}
%%
[aeiouAEIOU] {vcount++;}
[a-z,A-Z] {ccount++;}
%%
int yywrap(){}
int main()
{
printf("enter the string with vowels and consonants:");
yylex();
printf("\n no of vowels ::%d \n",vcount);
printf("\n no of consonants ::%d \n",ccount);
}
```

Output:



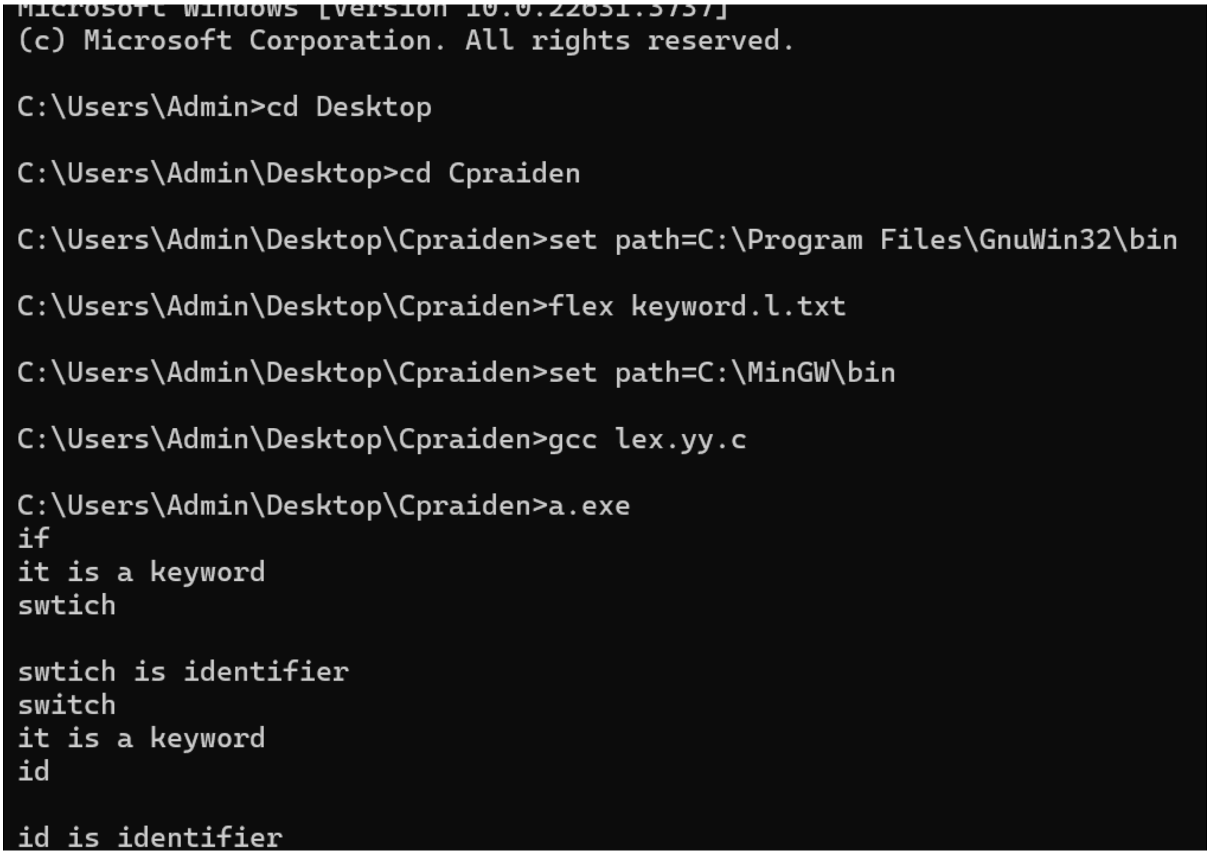
```
Microsoft Windows [Version 10.0.22631.3737]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin>cd Desktop

C:\Users\Admin\Desktop>cd Cpravowels

C:\Users\Admin\Desktop\Cpravowels>set path=C:\Program Files\GnuWin32\bin

C:\Users\Admin\Desktop\Cpravowels>flex vowels.l.txt

C:\Users\Admin\Desktop\Cpravowels>set path=C:\MinGW\bin

C:\Users\Admin\Desktop\Cpravowels>gcc lex.yy.c

C:\Users\Admin\Desktop\Cpravowels>a.exe
enter the string with vowels and consonants:bhanu teja

^Z

 no of vowels ::4

 no of consonants ::5

C:\Users\Admin\Desktop\Cpravowels>
```

5.write a LEX program to separate keywords and identifiers.

Program:

```
 %{
#include<stdio.h>
%}
%%
if|else|while|int|switch|for|char { printf("\n%s is a
KEYWORD", yytext);}
[a-zA-Z0-9]+ { printf("\n%s is IDENTIFIER", yytext);}
%%
int yywrap( ){}
int main()
{
yylex();
}
```

Output:



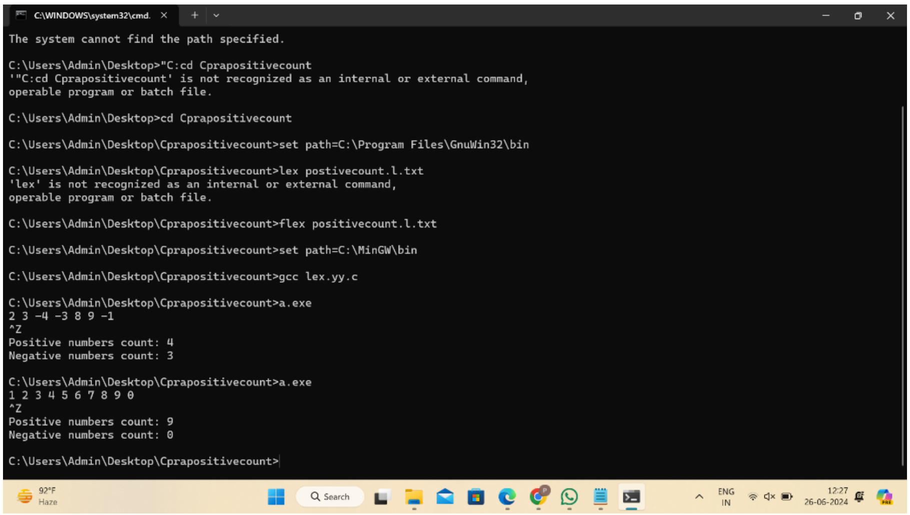
```
Microsoft Windows [Version 10.0.22631.3737]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin>cd Desktop

C:\Users\Admin\Desktop>cd Cpraiden

C:\Users\Admin\Desktop\Cpraiden>set path=C:\Program Files\GnuWin32\bin

C:\Users\Admin\Desktop\Cpraiden>flex keyword.l.txt

C:\Users\Admin\Desktop\Cpraiden>set path=C:\MinGW\bin

C:\Users\Admin\Desktop\Cpraiden>gcc lex.yy.c

C:\Users\Admin\Desktop\Cpraiden>a.exe
if
it is a keyword
swtich

swtich is identifier
switch
it is a keyword
id

id is identifier
```

6. Write a LEX program to identify and count positive and negative numbers.

Program:

```
%{
   int positive_count = 0;
   int negative_count = 0;
%}

DIGIT      [0-9]
SIGN       [-+]

%%
{SIGN}?{DIGIT}+      {
                       int num = atoi(yytext);
                       if (num > 0)
                           positive_count++;
                       else if (num < 0)
                           negative_count++;
                   }
.|\n               ; /* ignore any other characters */

%%

int yywrap() {
   return 1; // Indicate end of input
}

int main(int argc, char **argv) {
   yylex();
   printf("Positive numbers count: %d\n", positive_count);
   printf("Negative numbers count: %d\n", negative_count);
   return 0;
}
```

Output:



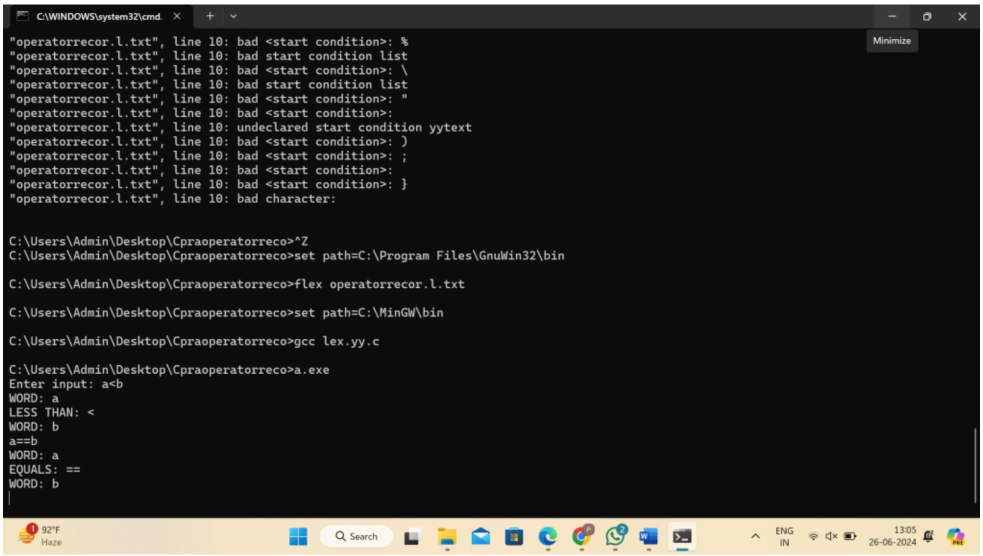
```
The system cannot find the path specified.

C:\Users\Admin\Desktop>"C:cd Cprapositivecount
'"C:cd Cprapositivecount' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\Admin\Desktop>cd Cprapositivecount

C:\Users\Admin\Desktop\Cprapositivecount>set path=C:\Program Files\GnuWin32\bin

C:\Users\Admin\Desktop\Cprapositivecount>lex postivecount.l.txt
'lex' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\Admin\Desktop\Cprapositivecount>flex positivecount.l.txt

C:\Users\Admin\Desktop\Cprapositivecount>set path=C:\MinGW\bin

C:\Users\Admin\Desktop\Cprapositivecount>gcc lex.yy.c

C:\Users\Admin\Desktop\Cprapositivecount>a.exe
2 3 -4 -3 8 9 -1
^Z
Positive numbers count: 4
Negative numbers count: 3

C:\Users\Admin\Desktop\Cprapositivecount>a.exe
1 2 3 4 5 6 7 8 9 0
^Z
Positive numbers count: 9
Negative numbers count: 0

C:\Users\Admin\Desktop\Cprapositivecount>
```

7. Write a LEX program to recognise operators and words in a statement.

Program:

```
%{
#include <stdio.h>
%}
%%
[a-zA-Z]+    { printf("WORD: %s\n", yytext); }
"=="         { printf("EQUALS: %s\n", yytext); }
"!="         { printf("NOT EQUALS: %s\n", yytext); }
"<="         { printf("LESS THAN OR EQUAL: %s\n", yytext); }
">="         { printf("GREATER THAN OR EQUAL: %s\n", yytext); }
"<"          { printf("LESS THAN: %s\n", yytext); }
">"          { printf("GREATER THAN: %s\n", yytext); }
[ \t\n]+     { /* Ignore whitespace */ }
.            { printf("UNKNOWN: %s\n", yytext); }
%%

int main() {
    while (1) {
        printf("Enter input: ");
        yylex();
    }
    return 0;
}

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

Output:



```
"operatorrecor.l.txt", line 10: bad <start condition>: %
"operatorrecor.l.txt", line 10: bad start condition list
"operatorrecor.l.txt", line 10: bad <start condition>: \
"operatorrecor.l.txt", line 10: bad start condition list
"operatorrecor.l.txt", line 10: bad <start condition>: "
"operatorrecor.l.txt", line 10: bad <start condition>:
"operatorrecor.l.txt", line 10: undeclared start condition yytext
"operatorrecor.l.txt", line 10: bad <start condition>: )
"operatorrecor.l.txt", line 10: bad <start condition>: ;
"operatorrecor.l.txt", line 10: bad <start condition>:
"operatorrecor.l.txt", line 10: bad <start condition>: }
"operatorrecor.l.txt", line 10: bad character:


C:\Users\Admin\Desktop\Cpraoperatorreco>^Z
C:\Users\Admin\Desktop\Cpraoperatorreco>set path=C:\Program Files\GnuWin32\bin

C:\Users\Admin\Desktop\Cpraoperatorreco>flex operatorrecor.l.txt

C:\Users\Admin\Desktop\Cpraoperatorreco>set path=C:\MinGW\bin

C:\Users\Admin\Desktop\Cpraoperatorreco>gcc lex.yy.c

C:\Users\Admin\Desktop\Cpraoperatorreco>a.exe
Enter input: a<b
WORD: a
LESS THAN: <
WORD: b
a==b
WORD: a
EQUALS: ==
WORD: b
```

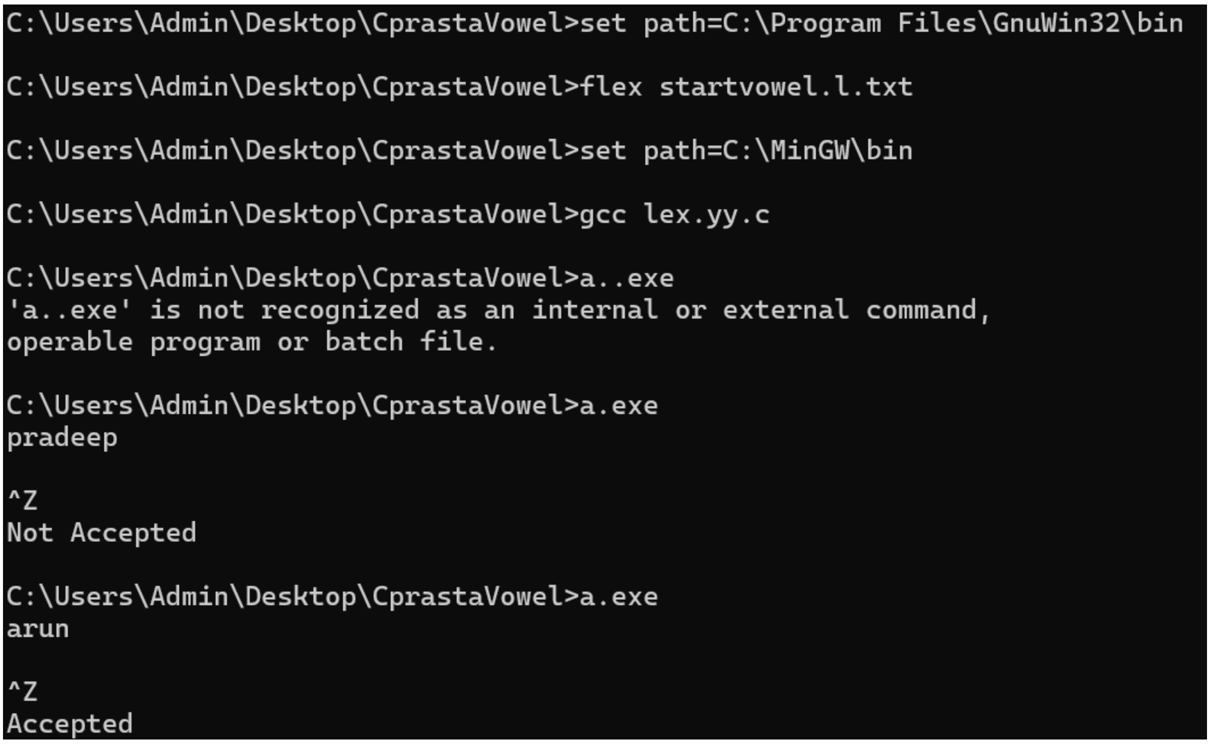
8. Write a LEX program to accept string starting with vowel.

Program:

```
%{
#include <stdio.h>

int flag = 0;
%}

%%
[aeiouAEIOU][a-zA-Z0-9]+    { flag = 1; }
[a-zA-Z0-9]+                { /* no action needed */ }
%%
```

```
int main() {
    yylex();
    if (flag == 1) {
        printf("Accepted\n");
    } else {
        printf("Not Accepted\n");
    }
    return 0;
}

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

Output:



```
C:\Users\Admin\Desktop\CprastaVowel>set path=C:\Program Files\GnuWin32\bin

C:\Users\Admin\Desktop\CprastaVowel>flex startvowel.l.txt

C:\Users\Admin\Desktop\CprastaVowel>set path=C:\MinGW\bin

C:\Users\Admin\Desktop\CprastaVowel>gcc lex.yy.c

C:\Users\Admin\Desktop\CprastaVowel>a..exe
'a..exe' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\Admin\Desktop\CprastaVowel>a.exe
pradeep

^Z
Not Accepted

C:\Users\Admin\Desktop\CprastaVowel>a.exe
arun

^Z
Accepted
```