



**Ganpat
University**

॥ विद्यया समाजोत्कर्षः ॥

**Institute of
Computer
Technology**

Name: Tushar Panchal

En.No: 21162101014

Sub: CD(Compiler Design)

Branch: CBA

Batch:71

-----PRACTICAL 02-----

1)lex program to count number of words and digit

2)Write a lex Program to Scan and Count the number of characters, words, digits, vowels, consonant, special characters and lines in a file.

Make Word.l file in which we will put our code

```
1  /** Definition Section has one variable
2  which can be accessed inside yylex()
3  and main() */
4  %{
5  int words = 0,num=0;
6  %}
7
8  %%
9  [a-zA-Z]+ {printf("%s is word\n", yytext);
10             words++;}
11
12  [0-9]+ {printf("%s is number\n", yytext);
13          num++;}
14
15  [ \n\t] {printf("%s is space\n", yytext); }
16          {printf("%s not valid token\n", yytext);}
17  .
18  {return 0;}
19  %}
20
21  int yywrap(){
22
23  int main(){
24
25  yyin=fopen("a.txt","r");
26  yylex();
27
28  printf("\nwords = %d Number = %d\n", words,num);
29
30  return 0;
31  }
32
```

Run that file

```
>_ pwsh 2 25ms
>> flex .\word.l
".\word.l", line 18: warning, rule cannot be matched
```

Compile that lex.yy.c file which is generated.

```
>_ pwsh 2 2ms
>> gcc .\lex.yy.c
>_ pwsh 2 970ms
>>
```

Now run execution file "a.out"

```
>_ pwsh 2 970ms
>> .\a.exe
Tushar Panchal 123456789
Tushar is word
    is space
Panchal is word
    is space
123456789 is number

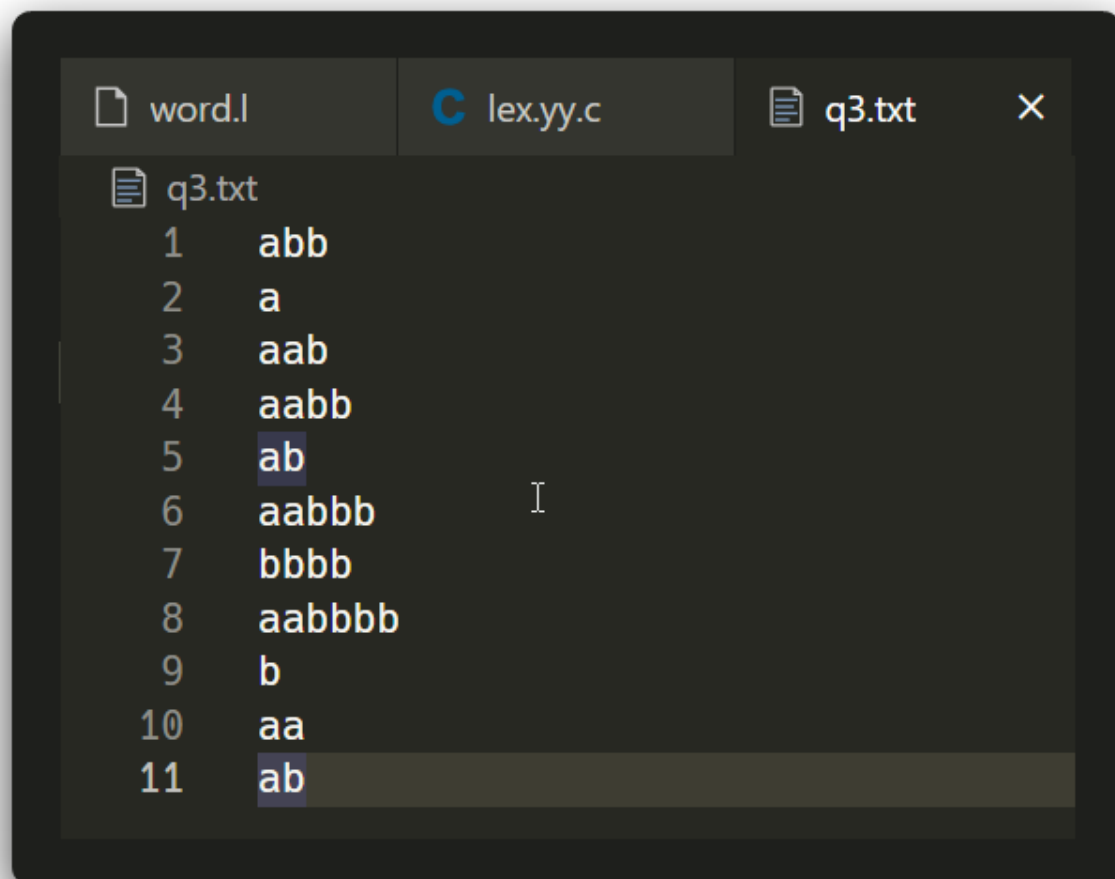
    is space

    is space

    is space
^Z^Z
>_ pwsh 2 44s 247ms
>>
```

3) Write a lex Program to recognize regular expression under 'a', 'a*b+', 'abb', b* over the input set {a,b}.

abb
a
aab
aabb
ab
aabbb
bbbb
aabbbb
b
aa
ab



The screenshot shows a text editor window with three tabs: 'word.l', 'lex.yy.c', and 'q3.txt'. The 'q3.txt' tab is active, displaying the following content:

```

1  abb
2  a
3  aab
4  aabb
5  ab
6  aabbb
7  bbbb
8  aabbbb
9  b
10 aa
11 ab
  
```

Task3.l:

```

%{
#include <stdio.h>
#include <string.h>
int count_a = 0;
int count_a_star_b_plus = 0;
  
```

```

int count_abb = 0;
int count_b_star = 0;
extern FILE *yyin;
%}

%%

abb          { printf("Matched 'abb'\n"); count_abb++; }
a*b+        { printf("Matched 'a*b+'\n"); count_a_star_b_plus++; }
a           { printf("Matched 'a'\n"); count_a++; }
b+          { /* We use 'b+' here, but will adjust count for 'b*'
later */ }
.           { /* Ignore any other characters */ }

%%

int yywrap() {
    return 1;
}

int main() {
    FILE *file = fopen("q3.txt", "r");
    if (!file) {
        perror("Error opening file");
        return 1;
    }

    yyin = file;
    yylex();

    rewind(file);
    yyin = file;

    char buffer[1024];
    while (fgets(buffer, sizeof(buffer), file)) {
        char *ptr = buffer;
        while (*ptr) {
            if (*ptr == 'b') {
                count_b_star++;
                while (*ptr == 'b') ptr++; // Skip over the rest of the
'b's
            } else {
                ptr++;
            }
        }
    }

    fclose(file);

```

```
printf("\nCounts:\n'a': %d\n'a*b+': %d\n'abb': %d\n'b*': %d\n",
count_a, count_a_star_b_plus, count_abb, count_b_star);
return 0;
}
```

```
>_ pwsh 2 1ms
>> flex .\task3.l
".\task3.l", line 16: warning, rule cannot be matched
>_ pwsh 2 23ms
>> gcc .\lex.yy.c
```

```
>_ pwsh 2 152ms
>> .\a.exe
Matched 'abb'

Matched 'a'

Matched 'a*b+'

Matched 'a*b+'

Matched 'a*b+'

Matched 'a*b+'

Matched 'a*b+'

Matched 'a*b+'

Matched 'a*b+'

Matched 'a'

Matched 'a'

Matched 'a*b+'

Counts:
'a': 3
'a*b+': 8
'abb': 1
'b*': 9
```