Name – Devansh

SAP – 500107123

Batch – B1

Experiment 2 Compiler design - Introductory Problems using Lex Tool

Q1 - WAP to count number of vowels and consonants in a given string.

File Name - vowels.l

```
%{
#include <stdio.h>
int vowels = 0, consonants = 0;
int yywrap(void) { return 1; }
%}
%%
[aeiouAEIOU] { vowels++; }
[a-zA-Z] { consonants++; }
.|\n { /* Ignore other characters */ }
%%

int main() {
   printf("Enter a string: ");
   yylex();
   printf("Number of vowels: %d\n", vowels);
   printf("Number of consonants: %d\n", consonants);
   return 0;
}
```

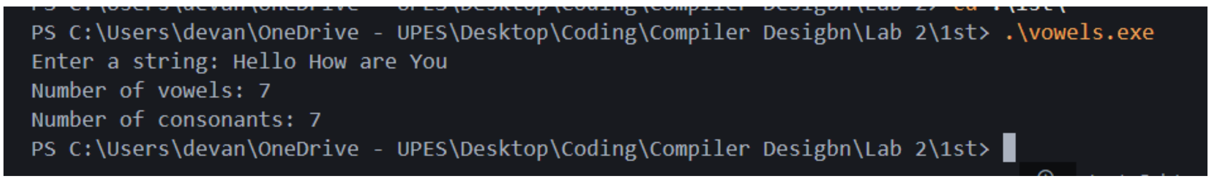
Run Commands –

Flex vowels.l

gcc lex.yy.c -o vowels

.\vowels.exe

Output –



```
PS C:\Users\devan\OneDrive - UPES\Desktop\Coding\Compiler Desigbn\Lab 2\1st> .\vowels.exe
Enter a string: Hello How are You
Number of vowels: 7
Number of consonants: 7
PS C:\Users\devan\OneDrive - UPES\Desktop\Coding\Compiler Desigbn\Lab 2\1st>
```

Q2 - WAP to count the number of characters, words, spaces, and end of lines in a given input file.

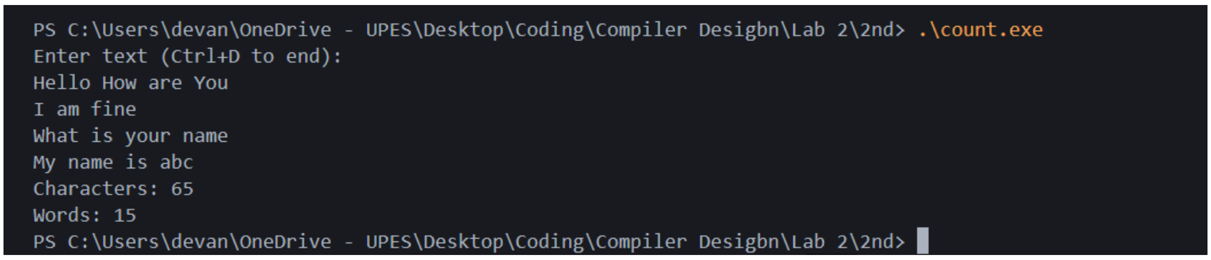
File name- count.l

```
%{
#include <stdio.h>
#include <ctype.h> // for isspace
int chars = 0, words = 0, spaces = 0, lines = 0;
int yywrap(void) { return 1; }
%}
%%
[ \t] { spaces++; chars++; }  // count spaces and tabs as chars
\n { lines++; chars++; }      // count newlines as chars
[^\t\n ]+ { words++; chars += yyleng; }  // count words and characters
. { chars++; }  // count all other characters
%%
int main() {
    printf("Enter text (Ctrl+D to end):\n");
    yylex();
    printf("Characters: %d\n", chars);
    printf("Words: %d\n", words);
    printf("Spaces: %d\n", spaces);
    printf("Lines: %d\n", lines);
    return 0;
}
```

Run Commands –

```
flex count.l
gcc lex.yy.c -o count
.\count.exe
```

Output



```
PS C:\Users\devan\OneDrive - UPES\Desktop\Coding\Compiler Desigbn\Lab 2\2nd> .\count.exe
Enter text (Ctrl+D to end):
Hello How are You
I am fine
What is your name
My name is abc
Characters: 65
Words: 15
PS C:\Users\devan\OneDrive - UPES\Desktop\Coding\Compiler Desigbn\Lab 2\2nd>
```

Q3 – WAP to count number of comment lines in a given C program.

File name – Comment.l

```
%{
#include <stdio.h>
int comment_lines = 0;
int yywrap(void) { return 1; }
%}
%%
```

```
\/\/.* { comment_lines++; } // Single-line comment

\/\*([^*]|\*+[^\/])*\*+\/ { comment_lines++; } // Multi-line comment

.|\n { /* Ignore other lines */ }

%%

int main() {

    printf("Enter the C program code (Ctrl+D to end):\n");

    yylex();

    printf("Number of comment lines: %d\n", comment_lines);

    return 0;

}
```

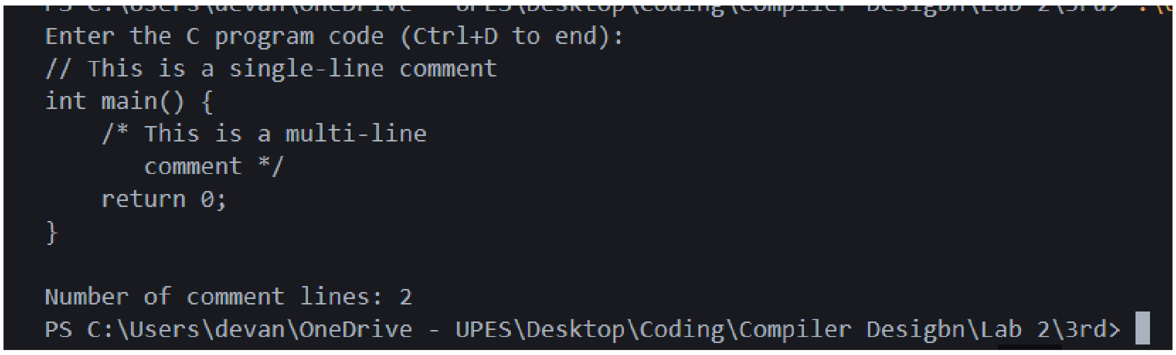
Run Commands –

```
Flex comment.l

gcc lex.yy.c -o comment

.\comment.exe
```

Output –



```
PS C:\Users\devan\OneDrive - UPES\Desktop\Coding\Compiler Desigbn\Lab 2\3rd> .\c
Enter the C program code (Ctrl+D to end):
// This is a single-line comment
int main() {
    /* This is a multi-line
       comment */
    return 0;
}

Number of comment lines: 2
PS C:\Users\devan\OneDrive - UPES\Desktop\Coding\Compiler Desigbn\Lab 2\3rd>
```