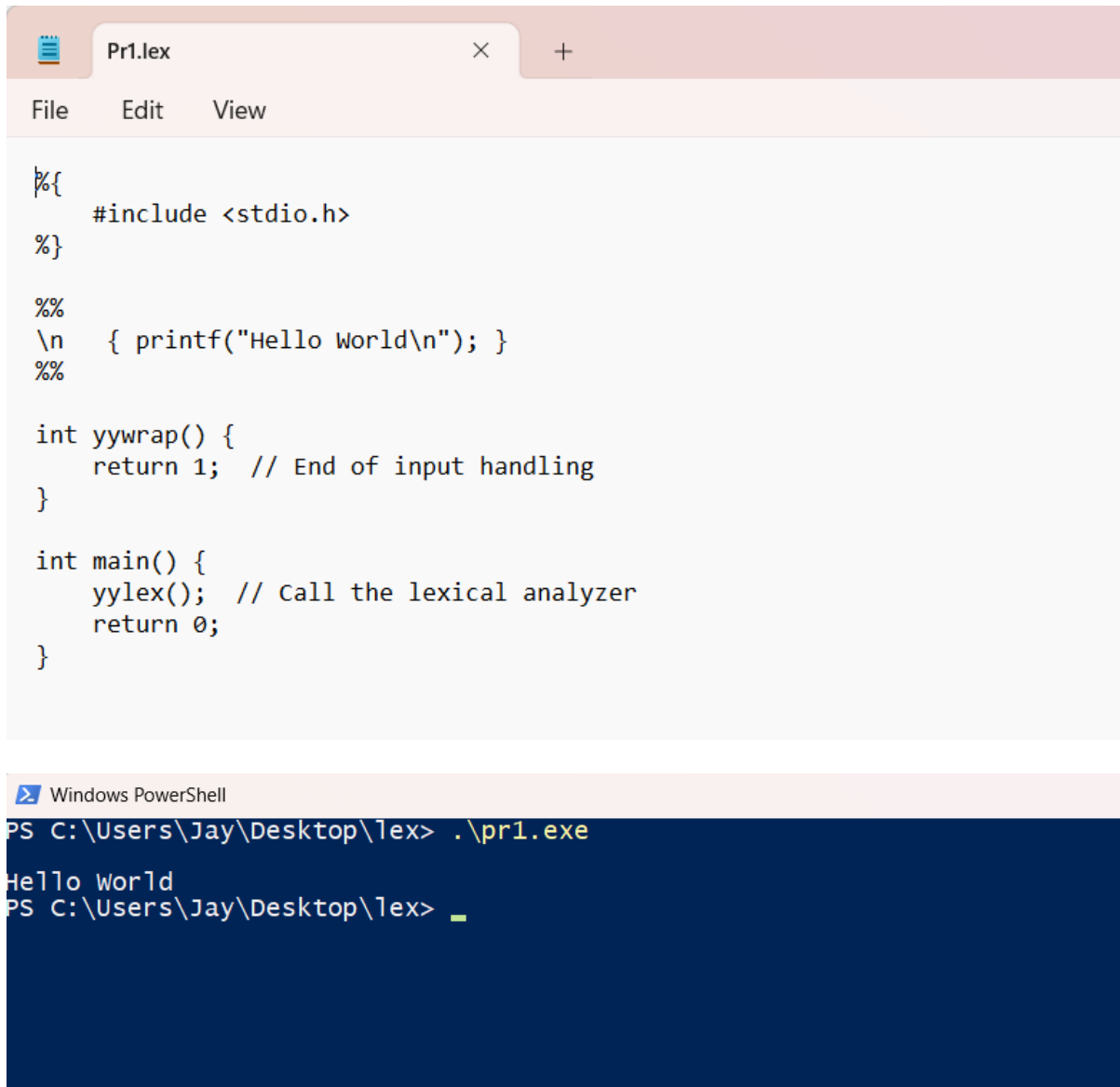


Practical – 2

Implement following programs using Lex.

Q1. Write a lex program to print hello world



The image shows a screenshot of a Lex program and its execution. The top part is a code editor window titled 'Pr1.lex' with a menu bar (File, Edit, View). The code is as follows:

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

%%
\n  { printf("Hello World\n"); }
%%

int yywrap() {
    return 1; // End of input handling
}

int main() {
    yylex(); // Call the lexical analyzer
    return 0;
}
```

The bottom part is a Windows PowerShell terminal window. It shows the command `PS C:\Users\Jay\Desktop\lex> .\pr1.exe` being executed, which results in the output `Hello world`. The prompt then returns to `PS C:\Users\Jay\Desktop\lex> _`.

Q2. Write a lex program to recognize tokens

```
PR2.lex
File Edit View

%{
    #include <stdio.h>
    #include <string.h>
}%

%%

int          { printf("Found an integer: %s\n", yytext); }
[a-zA-Z_][a-zA-Z0-9_]* { printf("Found an identifier: %s\n", yytext); }
[0-9]+       { printf("Found a number: %s\n", yytext); }
"+"|"-"|"*"|"/" { printf("Found an operator: %s\n", yytext); }
"="          { printf("Found an assignment operator: %s\n", yytext); }
"("          { printf("Found an opening parenthesis: %s\n", yytext); }
")"          { printf("Found a closing parenthesis: %s\n", yytext); }
";"          { printf("Found a semicolon: %s\n", yytext); }

%%

int yywrap() {
    return 1; // End of input handling
}

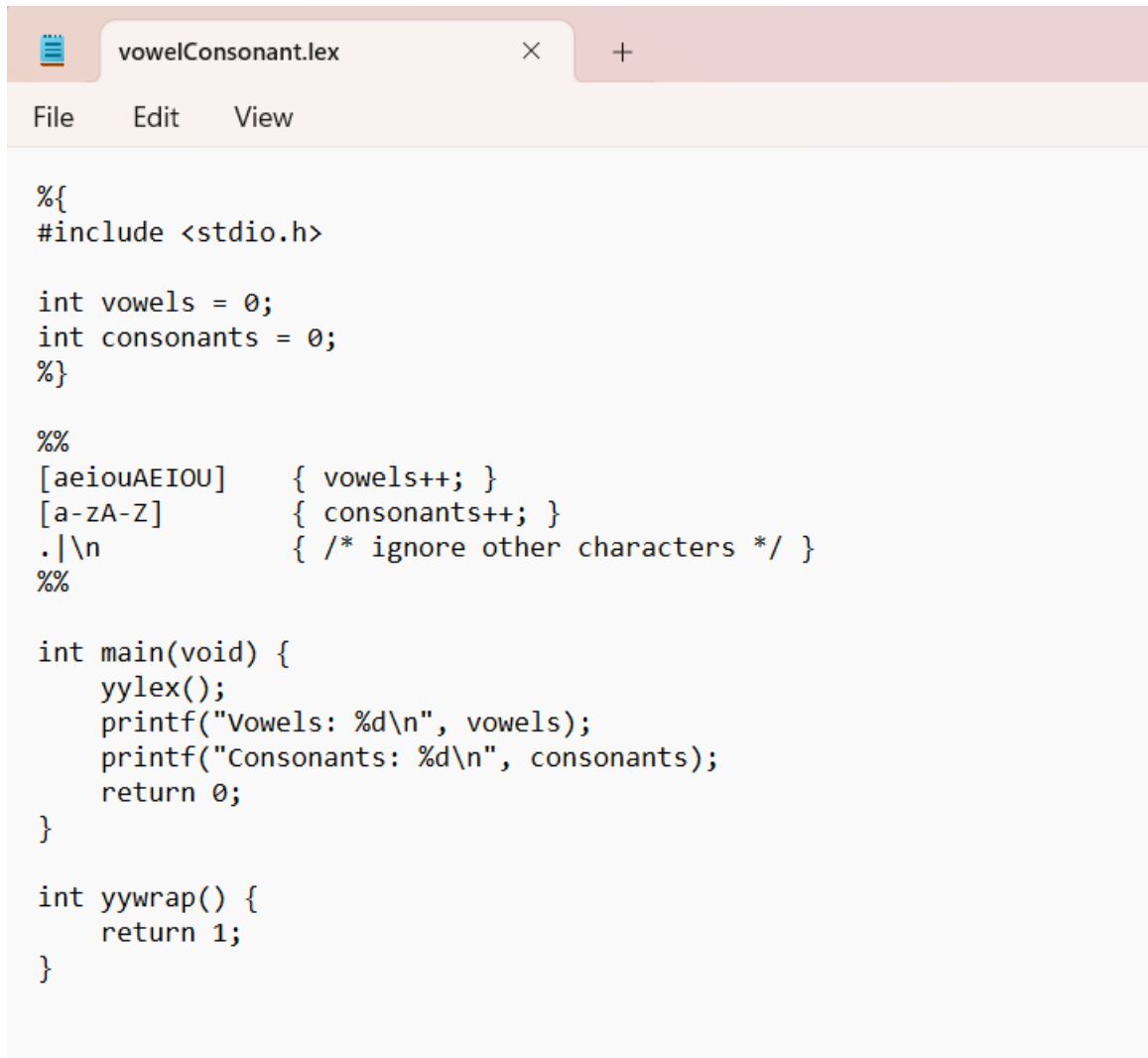
int main() {
    yylex(); // Call the lexical analyzer
    return 0;
}
```

```
PS C:\Users\Jay\Desktop\lex> .\Pr2.exe
Hello
Found an identifier: Hello

helo
Found an identifier: helo

abc
Found an identifier: abc
```

Q3. Write a lex program to count vowels and consonants



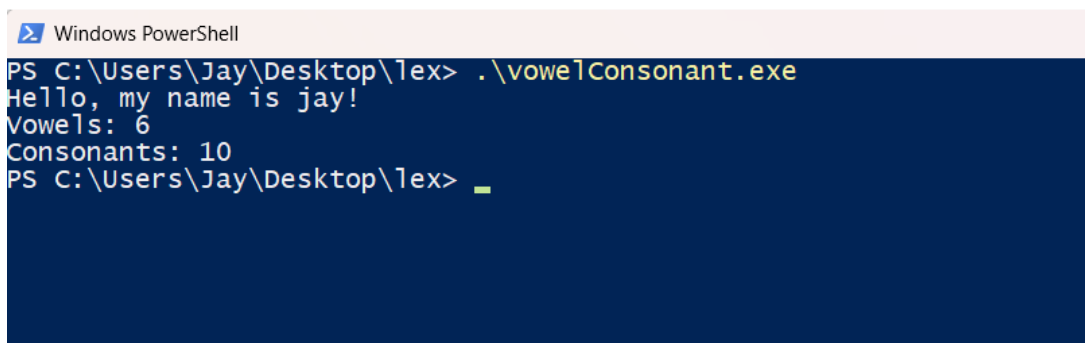
```
%{
#include <stdio.h>

int vowels = 0;
int consonants = 0;
}%

%%
[aeiouAEIOU]    { vowels++; }
[a-zA-Z]        { consonants++; }
.|\\n           { /* ignore other characters */ }
%%

int main(void) {
    yylex();
    printf("Vowels: %d\\n", vowels);
    printf("Consonants: %d\\n", consonants);
    return 0;
}

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



```
Windows PowerShell
PS C:\Users\Jay\Desktop\lex> .\vowelConsonant.exe
Hello, my name is jay!
Vowels: 6
Consonants: 10
PS C:\Users\Jay\Desktop\lex> _
```

Q4. Create a Lexer to take input from text file and count no of characters, no. of lines & no. of words.

```
wordcount.lex
File Edit View

%{
#include <stdio.h>

int char_count = 0;
int word_count = 0;
int line_count = 0;
%}

%%

.          { char_count++; }                // Count every character
[ \t]+     { /* Skip spaces/tabs, already counted as characters */ }
\n         { char_count++; line_count++; }    // Newline = new line, also a char
[A-Za-z0-9]+ { word_count++; char_count += yyleng; } // Word matched

%%

int main(int argc, char **argv)
{
    if (argc > 1) {
        FILE *fp = fopen(argv[1], "r");
        if (!fp) {
            perror("File opening failed");
            return 1;
        }
        yyin = fp;
    }
    yylex(); // Run the lexer

    printf("Number of characters: %d\n", char_count);
    printf("Number of words: %d\n", word_count);
    printf("Number of lines: %d\n", line_count);

    return 0;
}

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

```
PS C:\Users\Jay\Desktop\lex> .\wordcount.exe .\test.txt
Number of characters: 38
Number of words: 8
Number of lines: 0
PS C:\Users\Jay\Desktop\lex>
```