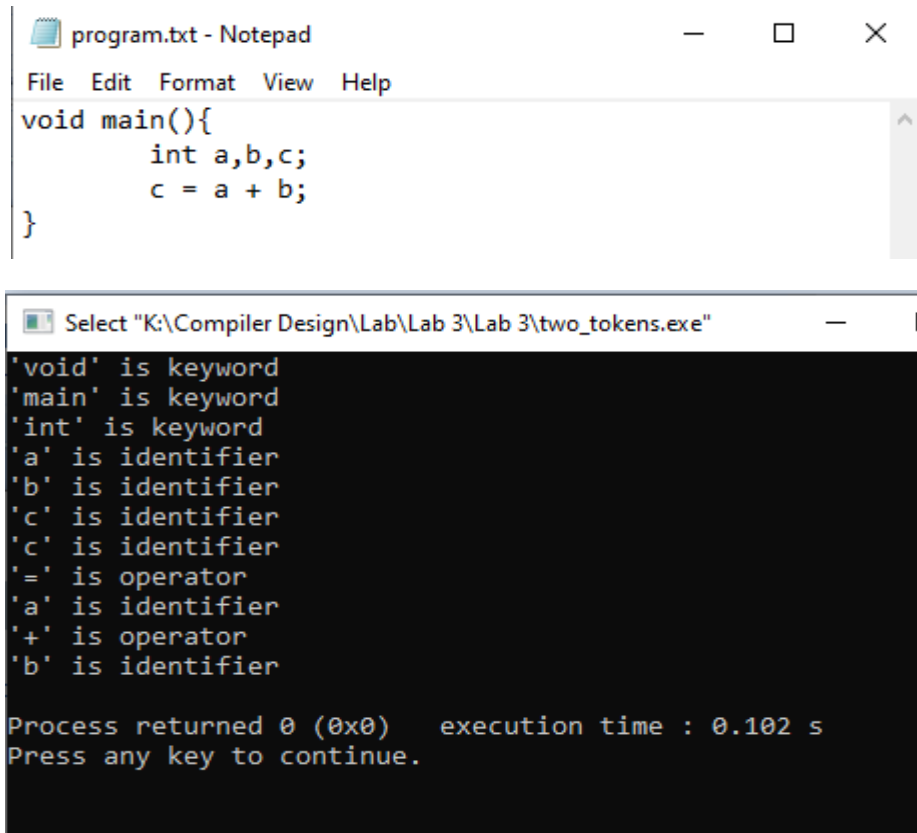


Lab Performance (02)

1. Write a C/C++ program that can read program from a file and detect all the tokens (e.g. valid or invalid identifier, keyword, operator) from the file.

For example:



The image shows two windows. The top window is a Notepad editor titled 'program.txt - Notepad' containing the following C code:

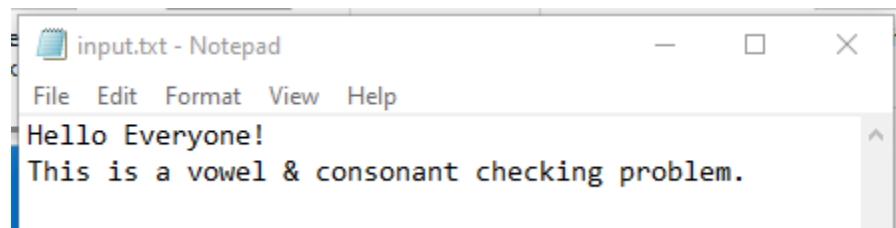
```
void main(){
    int a,b,c;
    c = a + b;
}
```

The bottom window is a console titled 'Select "K:\Compiler Design\Lab\Lab 3\Lab 3\two_tokens.exe"'. It displays the output of the program, identifying tokens in the code above:

```
'void' is keyword
'main' is keyword
'int' is keyword
'a' is identifier
'b' is identifier
'c' is identifier
'c' is identifier
'=' is operator
'a' is identifier
'+' is operator
'b' is identifier

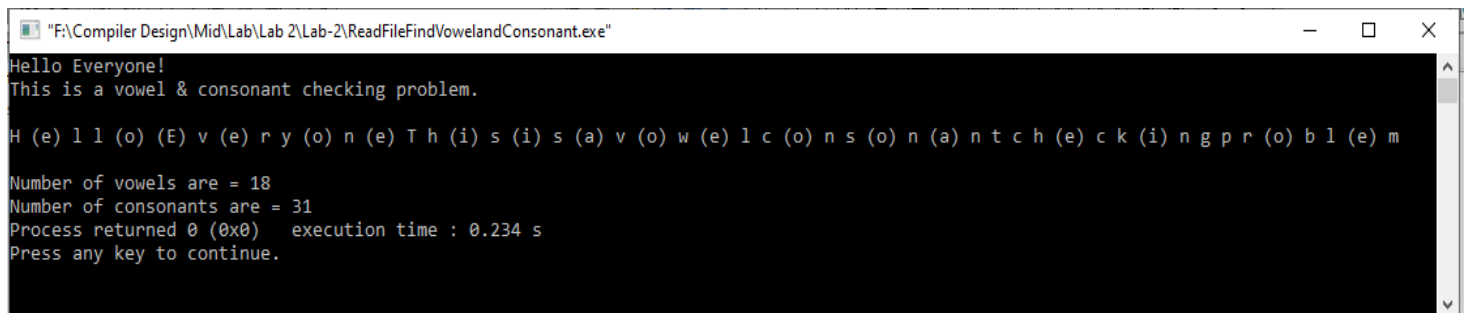
Process returned 0 (0x0)   execution time : 0.102 s
Press any key to continue.
```

2. Read a text message from a .txt file, calculate and print the number of vowels and consonants from the contents of that particular file in the console.



The image shows a Notepad editor titled 'input.txt - Notepad' containing the following text:

```
Hello Everyone!
This is a vowel & consonant checking problem.
```



The image shows a console window titled '"F:\Compiler Design\Mid\Lab\Lab 2\Lab 2\ReadFileFindVowelandConsonant.exe"'. It displays the output of the program, which reads the text from the file above and counts the vowels and consonants:

```
Hello Everyone!
This is a vowel & consonant checking problem.

H (e) l l (o) (E) v (e) r y (o) n (e) T h (i) s (i) s (a) v (o) w (e) l c (o) n s (o) n (a) n t c h (e) c k (i) n g p r (o) b l (e) m

Number of vowels are = 18
Number of consonants are = 31
Process returned 0 (0x0)   execution time : 0.234 s
Press any key to continue.
```

Lab Performance (02)

3. Write a C/C++ program to check if expression is correctly parenthesized or not.

For example:

```
Enter expression: a+(b*c)/e-(f+g*(h^i))/j

Expression is valid.

Process returned 0 (0x0)   execution time : 57.189 s
Press any key to continue.
```

4. Write a C/C++ program that can remove comment lines from a program (a task of preprocessor).

For example:

```
Before removing comment:

#include <stdio.h>

int main()
{
    printf("Hello World"); //This is a single line comment
    return 0;

    /* This is a
    multi line
    comment... */
}
v1.0;C:\Win

After removing comment:

#include <stdio.h>

int main()
{
    printf("Hello World");    return 0;
}
v1.0;C:\Win
Process returned 0 (0x0)   execution time : 0.101 s
Press any key to continue.
```