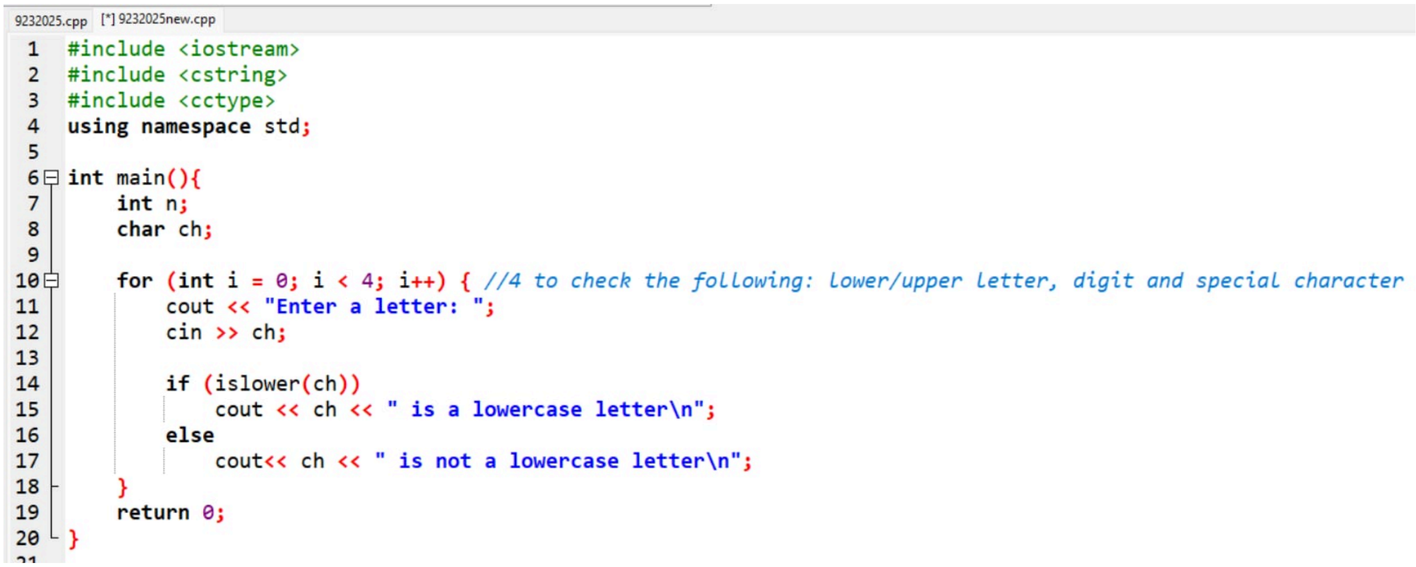
Hands-On Activity 4.4	
Characters and Strings	
Course Code: CPE007	**Program:** Computer Engineering
Course Title: Programming Logic and Design	**Date Performed:** September 23, 2025
Section: CPE11S1	**Date Submitted:** September 23, 2025
Name(s): Mendoza, Nathaniel B.	**Instructor: Engr. Jimlord M. Quejado**
6. Output	

1.

Code :

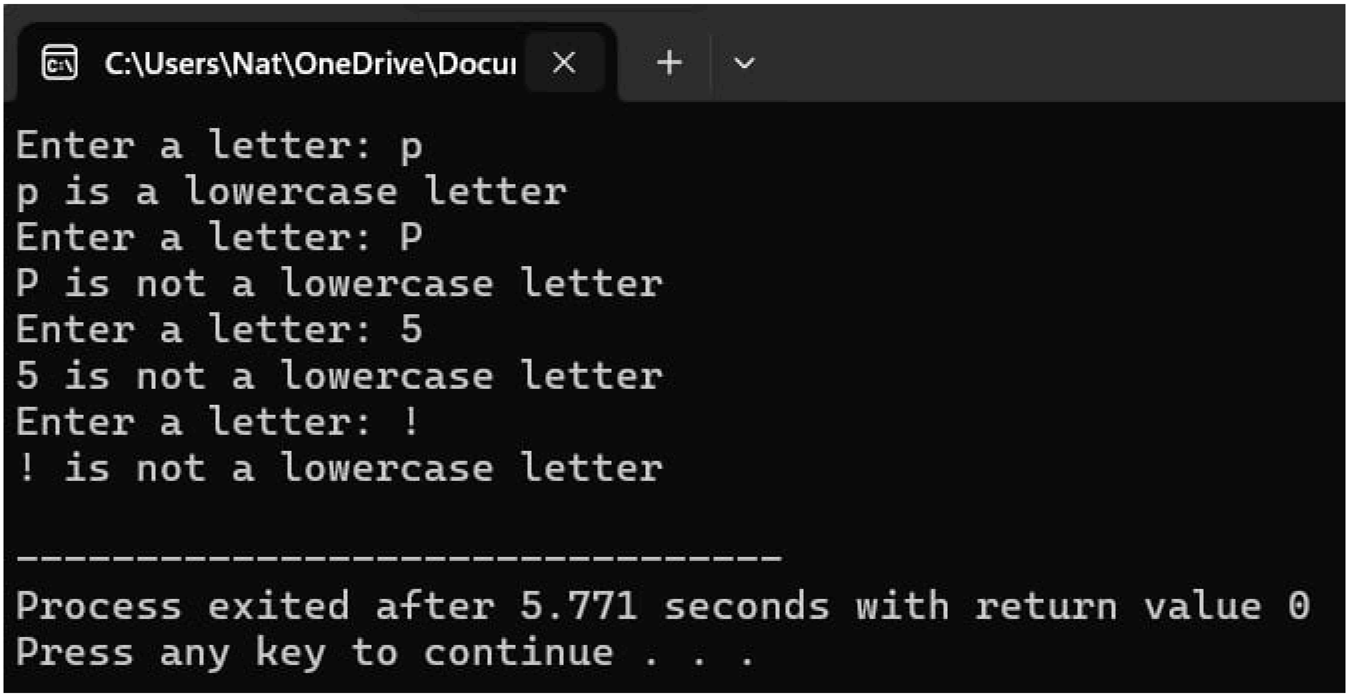


```
#include <iostream>
#include <cstring>
#include <cctype>
using namespace std;

int main(){
    int n;
    char ch;

    for (int i = 0; i < 4; i++) { //4 to check the following: lower/upper letter, digit and special character
        cout << "Enter a letter: ";
        cin >> ch;

        if (islower(ch))
            cout << ch << " is a lowercase letter\n";
        else
            cout<< ch << " is not a lowercase letter\n";
    }
    return 0;
}
```

Code Analysis :

I started the program with three header files: <iostream> which enables input and output operations such as cin and cout, then <cstring>, and <cctype> which gives access to character classification functions like islower(). The line using namespace std; is added so that the program can use standard library objects like cin and cout without needing to prefix them with std::. Then Inside the main() function, a character variable ch which is used to store each character entered by the user. The program then runs a for loop that repeats four times, as specified by the condition for (int i = 0; i < 4; i++) I used 4 so that I can implement 4 cases such as Upper and Lower cases then special characters and digits. Therefore the loop is designed to allow the user to test four different inputs. Within the loop, the program prompts the user with "Enter a letter: " and waits for input. Whatever character the user types is stored in the variable ch. The program then checks if the character is a lowercase letter using the function islower(ch). If the condition is true, the program outputs a message saying that the character is a lowercase letter. If the condition is false, it outputs a message saying that the character is not a lowercase letter.Finally, after the loop has executed four times, the program reaches return 0; which signals the successful termination of the program.

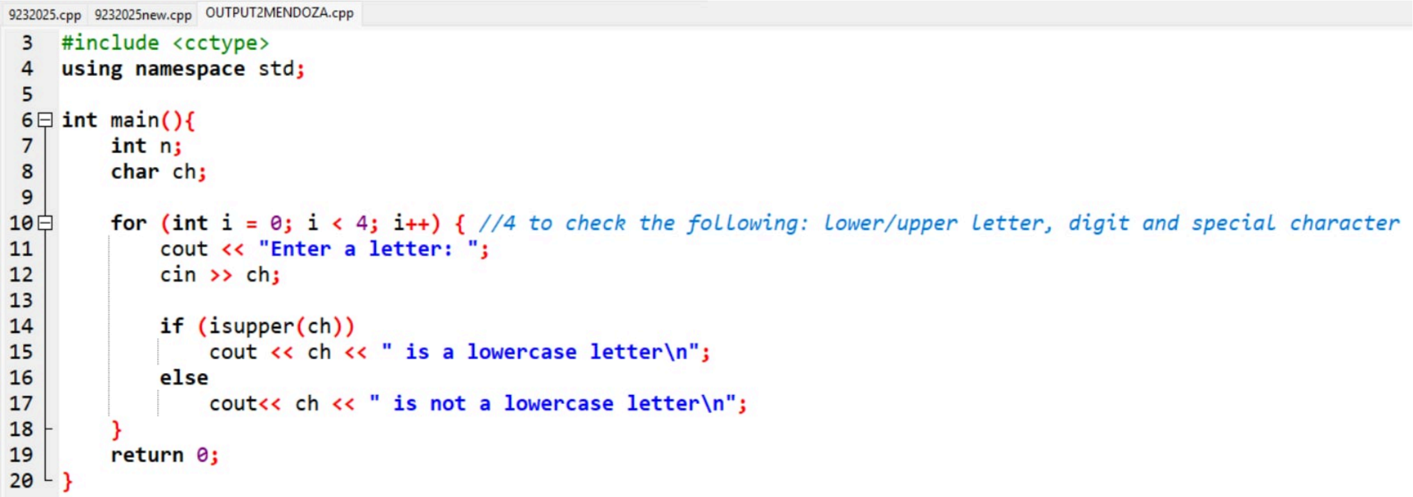
Code Output :



```
Enter a letter: p
p is a lowercase letter
Enter a letter: P
P is not a lowercase letter
Enter a letter: 5
5 is not a lowercase letter
Enter a letter: !
! is not a lowercase letter

--------------------------------
Process exited after 5.771 seconds with return value 0
Press any key to continue . . .
```

2.

Code :



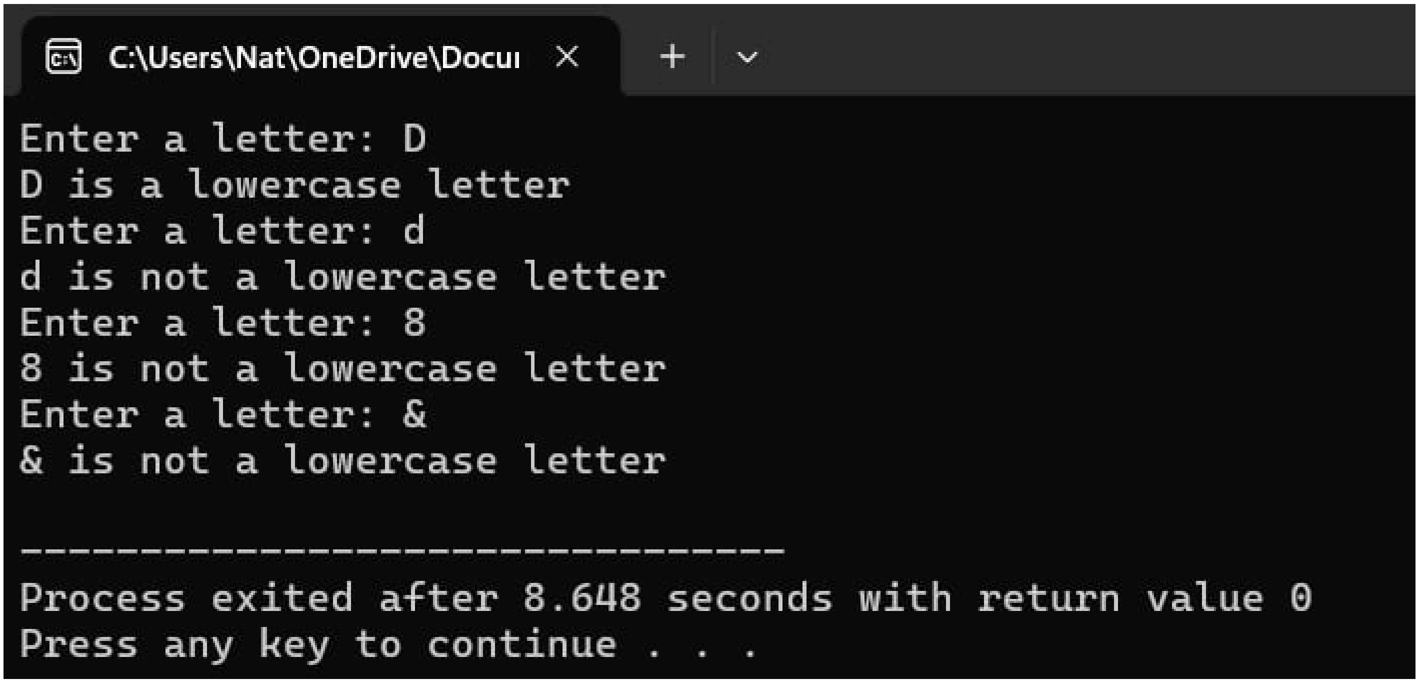
```cpp
#include <cctype>
using namespace std;

int main(){
    int n;
    char ch;

    for (int i = 0; i < 4; i++) { //4 to check the following: lower/upper letter, digit and special character
        cout << "Enter a letter: ";
        cin >> ch;

        if (isupper(ch))
            cout << ch << " is a lowercase letter\n";
        else
            cout<< ch << " is not a lowercase letter\n";
    }
    return 0;
}
```

Code Analysis :

I started the program with three header files: <iostream> which enables input and output operations such as cin and cout, then <cstring>, and <cctype> which gives access to character classification functions like isupper(). The line using namespace std; is added so that the program can use standard library objects like cin and cout without needing to prefix them with std::. Then Inside the main() function, a character variable ch which is used to store each character entered by the user. The program then runs a for loop that repeats four times, as specified by the condition for (int i = 0; i < 4; i++) I used 4 so that I can implement 4 cases such as Upper and Lower cases then special characters and digits. Therefore the loop is designed to allow the user to test four different inputs.

Within the loop, the program prompts the user with "Enter a letter: " and waits for input. Whatever character the user types is stored in the variable ch. The program then checks if the character is a lowercase letter using the function isupper(ch). If the condition is true, the program outputs a message saying that the character is a lowercase letter. If the condition is false, it outputs a message saying that the character is not a lowercase letter.Finally, after the loop has executed four times, the program reaches return 0; which signals the successful termination of the program.

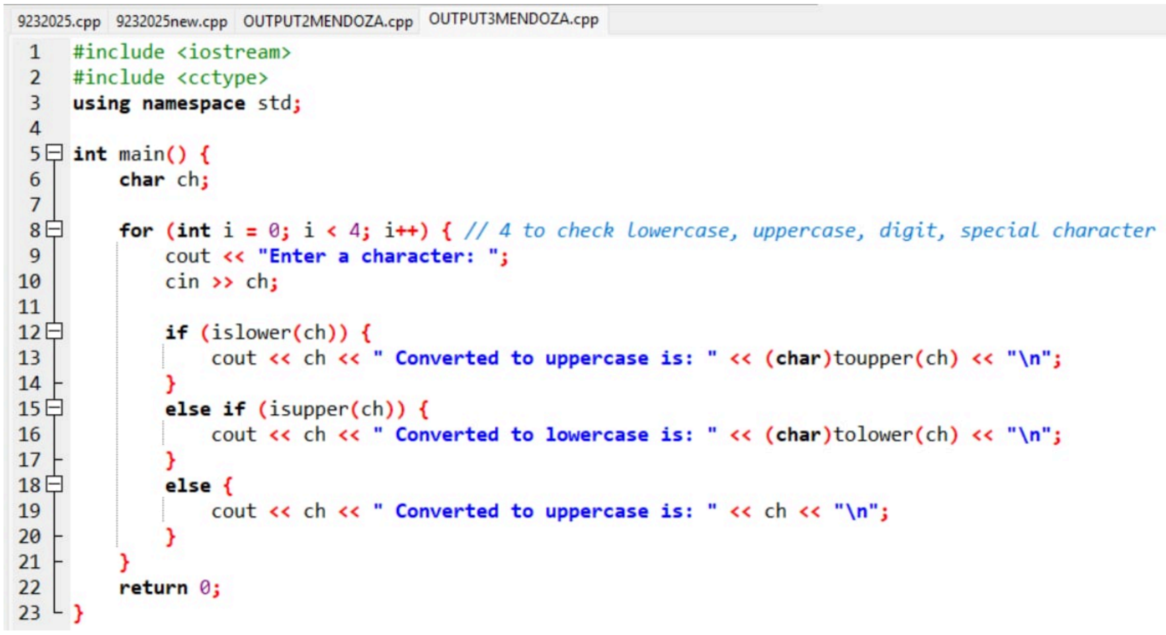
Code Output :



```
Enter a letter: D
D is a lowercase letter
Enter a letter: d
d is not a lowercase letter
Enter a letter: 8
8 is not a lowercase letter
Enter a letter: &
& is not a lowercase letter

--------------------------------
Process exited after 8.648 seconds with return value 0
Press any key to continue . . .
```

3.

Code :

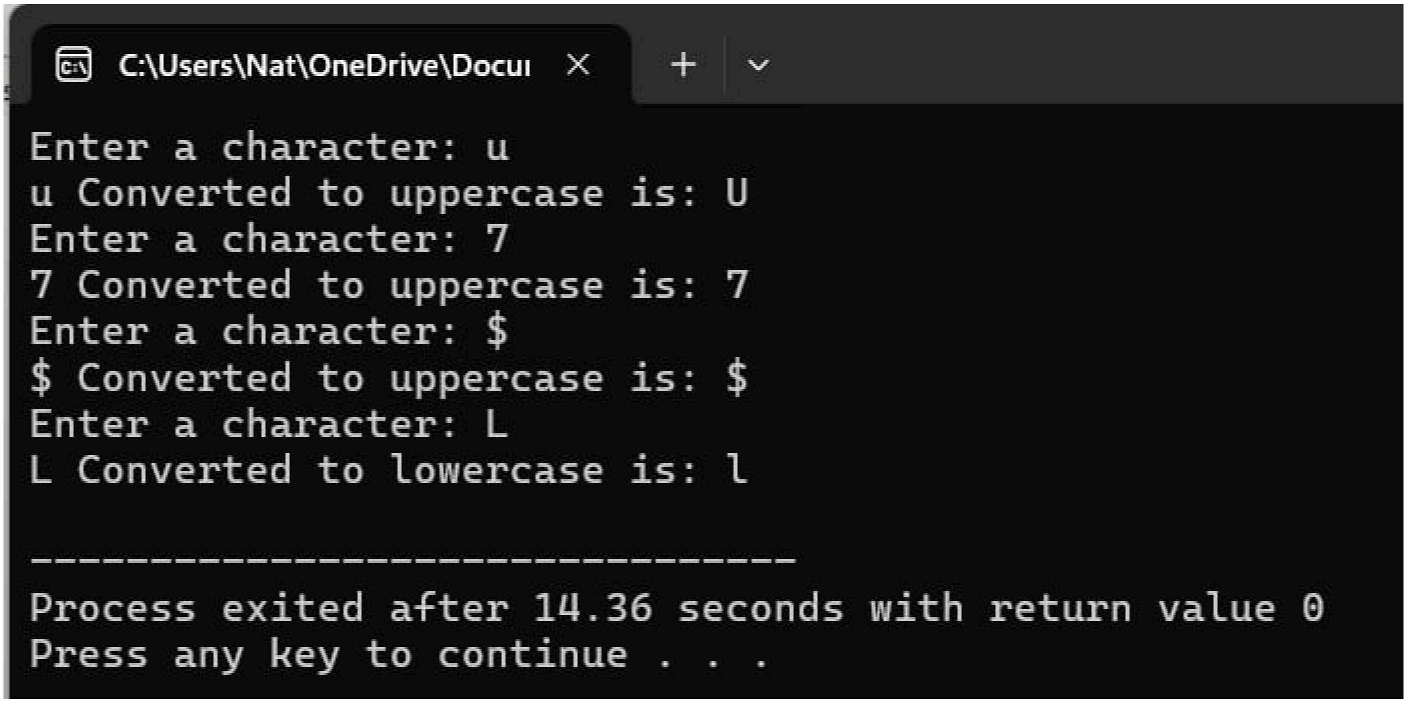


```cpp
#include <iostream>
#include <cctype>
using namespace std;

int main() {
    char ch;

    for (int i = 0; i < 4; i++) { // 4 to check lowercase, uppercase, digit, special character
        cout << "Enter a character: ";
        cin >> ch;

        if (islower(ch)) {
            cout << ch << " Converted to uppercase is: " << (char)toupper(ch) << "\n";
        }
        else if (isupper(ch)) {
            cout << ch << " Converted to lowercase is: " << (char)tolower(ch) << "\n";
        }
        else {
            cout << ch << " Converted to uppercase is: " << ch << "\n";
        }
    }
    return 0;
}
```

Code Analysis :

I started the program with the implementation of the <iostream> header, which allows the use of input and output functions like cin and cout. It also includes <cctype>, a library that provides useful character-handling functions such as islower(), isupper(), toupper(), and tolower(). The line using namespace std; is added so that the program can use standard objects without writing the prefix std:: Then inside main() function, a character variable ch is declared to hold each input entered by the user. A for loop is then created with the condition for (int i = 0; i < 4; i++), meaning the loop will run four times. This allows the user to test different types of characters such as lowercase letters, uppercase letters, digits, or special characters. Then the program will ask the user with "*Enter a character:* " and store the input in the variable ch. It then checks the type of character entered. If the input is a lowercase letter, as determined by islower(ch), the program uses toupper(ch) to convert it into its uppercase form and prints the result. If the input is an uppercase letter, as determined by isupper(ch), the program uses tolower(ch) to convert it into lowercase and displays it. If the input is neither lowercase nor uppercase (for example, a digit or a special character), the program simply outputs the same character without conversion. After the loop finishes running all four times, the program reaches return 0;, which indicates that it has executed successfully.

Code Output :

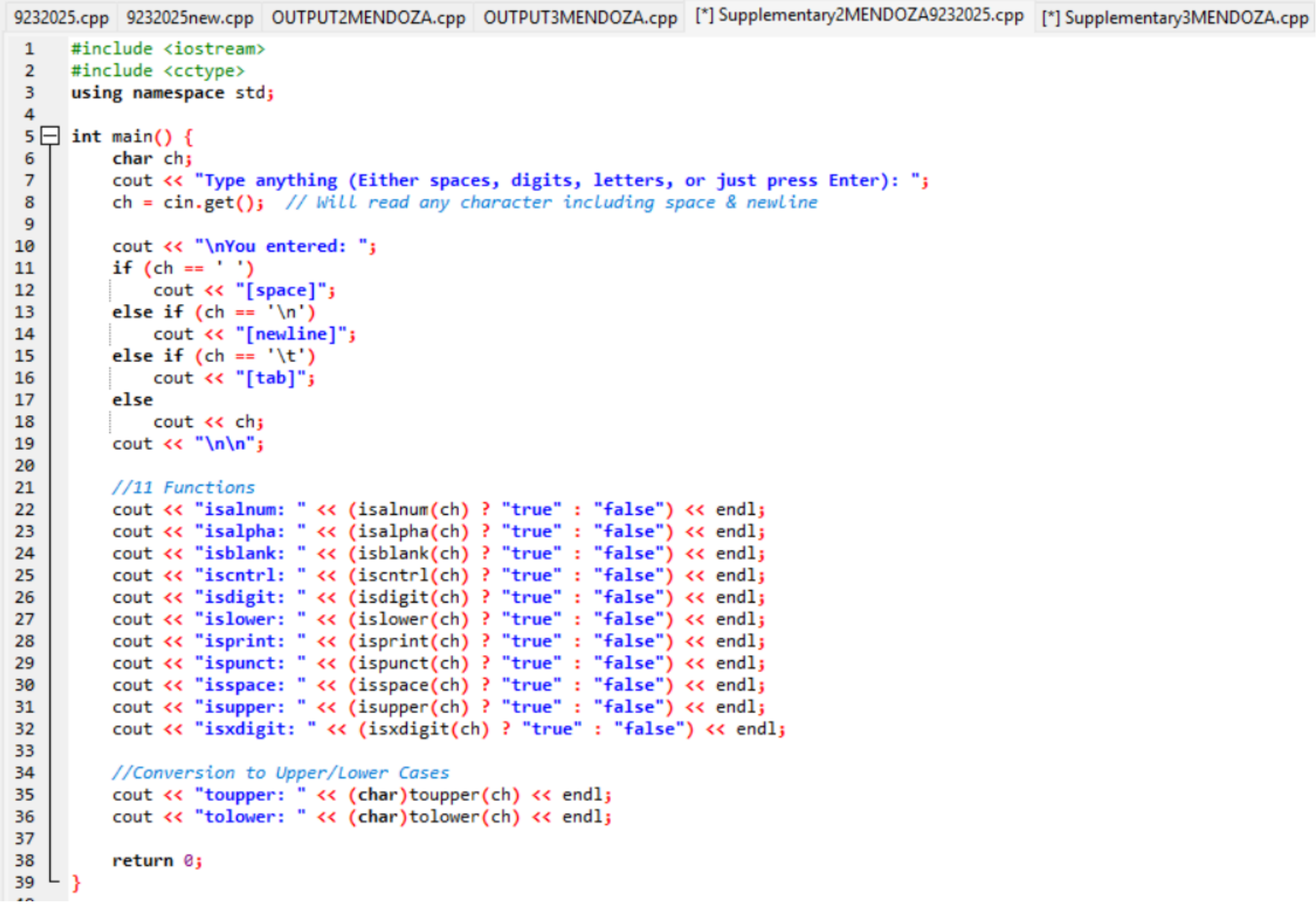


```
Enter a character: u
u Converted to uppercase is: U
Enter a character: 7
7 Converted to uppercase is: 7
Enter a character: $
$ Converted to uppercase is: $
Enter a character: L
L Converted to lowercase is: l

--------------------------------
Process exited after 14.36 seconds with return value 0
Press any key to continue . . .
```

7. Supplementary Activity

2.

Code :


9232025.cpp
9232025new.cpp
OUTPUT2MENDOZA.cpp
OUTPUT3MENDOZA.cpp
[*] Supplementary2MENDOZA9232025.cpp
[*] Supplementary3MENDOZA.cpp

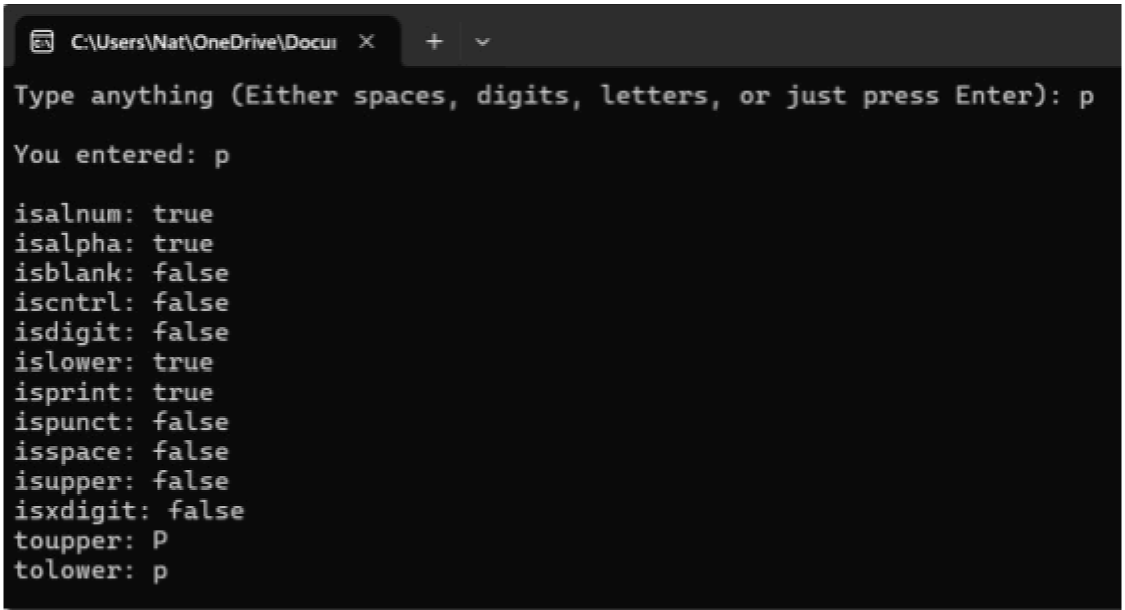
```cpp
#include <iostream>
#include <cctype>
using namespace std;

int main() {
    char ch;
    cout << "Type anything (Either spaces, digits, letters, or just press Enter): ";
    ch = cin.get();  // Will read any character including space & newline

    cout << "\nYou entered: ";
    if (ch == ' ')
        cout << "[space]";
    else if (ch == '\n')
        cout << "[newline]";
    else if (ch == '\t')
        cout << "[tab]";
    else
        cout << ch;
    cout << "\n\n";

    //11 Functions
    cout << "isalnum: " << (isalnum(ch) ? "true" : "false") << endl;
    cout << "isalpha: " << (isalpha(ch) ? "true" : "false") << endl;
    cout << "isblank: " << (isblank(ch) ? "true" : "false") << endl;
    cout << "iscntrl: " << (iscntrl(ch) ? "true" : "false") << endl;
    cout << "isdigit: " << (isdigit(ch) ? "true" : "false") << endl;
    cout << "islower: " << (islower(ch) ? "true" : "false") << endl;
    cout << "isprint: " << (isprint(ch) ? "true" : "false") << endl;
    cout << "ispunct: " << (ispunct(ch) ? "true" : "false") << endl;
    cout << "isspace: " << (isspace(ch) ? "true" : "false") << endl;
    cout << "isupper: " << (isupper(ch) ? "true" : "false") << endl;
    cout << "isxdigit: " << (isxdigit(ch) ? "true" : "false") << endl;

    //Conversion to Upper/Lower Cases
    cout << "toupper: " << (char)toupper(ch) << endl;
    cout << "tolower: " << (char)tolower(ch) << endl;

    return 0;
}
```

Code Analysis :

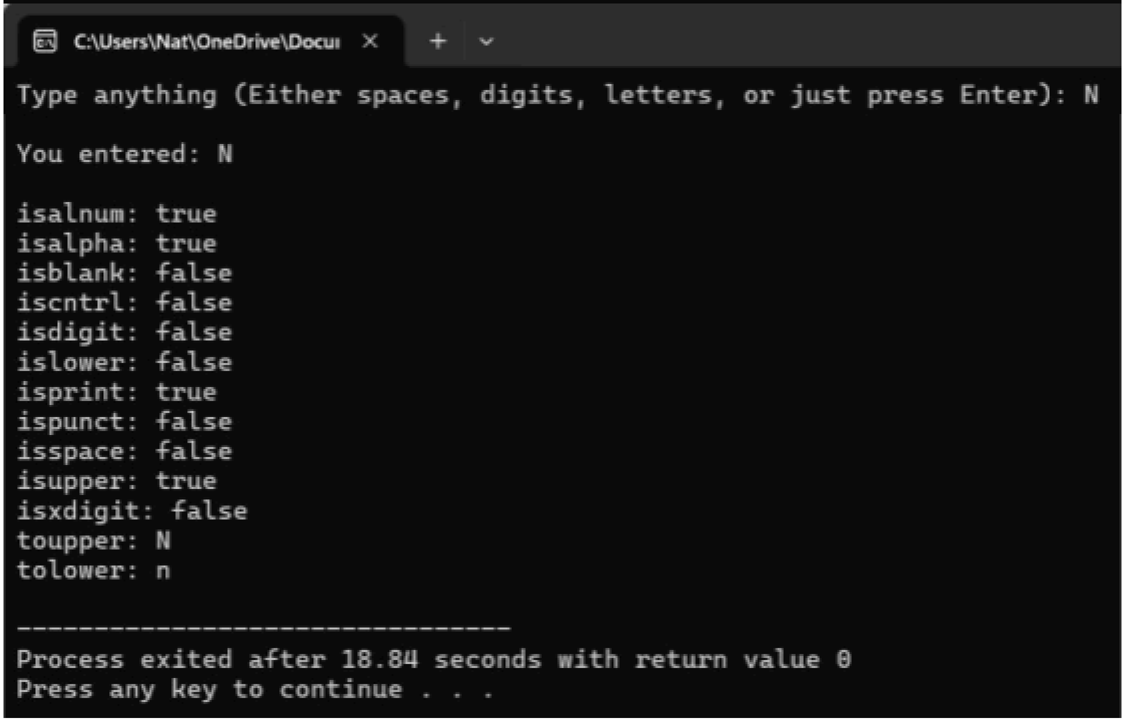
I started the program by including the header files such as <iostream> and <cctype>. The <iostream> header allows the program to use input and output commands such as cin and cout, while <cctype> provides several functions that are useful for checking and manipulating characters, like testing if a character is a digit, a letter, or converting it to uppercase or lowercase. Then the line using namespace std; is used so that I don't need to type std:: before every standard function. Inside the main() function, a character variable ch is declared. The program then displays a message asking the user to type basically anything, whether it is spaces, digits, letters, or even just pressing the Enter key. To properly capture all possible characters including spaces and newline characters, the program uses cin.get() instead of the normal cin >> ch. So that this ensures that the input character is read exactly as it is, without being skipped or ignored. After reading the character, the program checks what was entered and displays it. For example, if we look through the outputs of the code we have inputs such as space, newline, or tab, the program prints special labels like [space], [newline], or [tab]. Otherwise, it simply shows the character itself. Next, the program tests the character against eleven different <cctype> classification functions. These include isalnum() to check if the character is alphanumeric, isalpha() for alphabetic, isblank() for space or tab, iscntrl() for control characters, isdigit() for numbers, islower() for lowercase, isprint() for printable characters, ispunct() for punctuation, isspace() for whitespace characters, isupper() for uppercase, and isxdigit() for hexadecimal digits. Each of these checks outputs either "true" or "false" depending on the input. Finally, the program demonstrates character conversion by using toupper() to show the uppercase form of the input and tolower() to show the lowercase form of the input. This helps visualize how the same character can be transformed into different cases. After all the tests and conversions, the program ends with return 0; to indicate successful execution.

Code Output :

Lower and Upper cases of letter

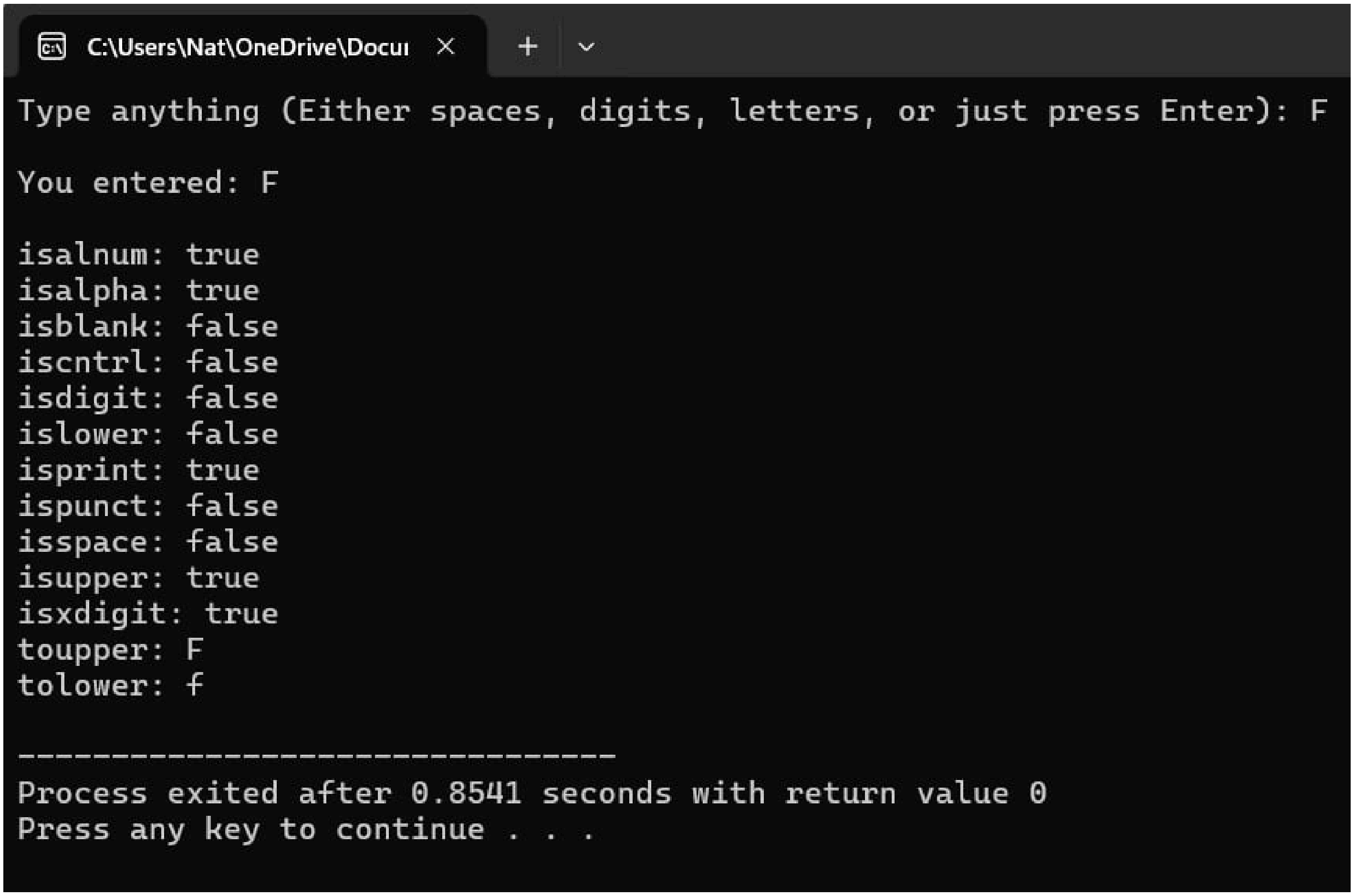


```
Type anything (Either spaces, digits, letters, or just press Enter): p

You entered: p

isalnum: true
isalpha: true
isblank: false
iscntrl: false
isdigit: false
islower: true
isprint: true
ispunct: false
isspace: false
isupper: false
isxdigit: false
toupper: P
tolower: p
```



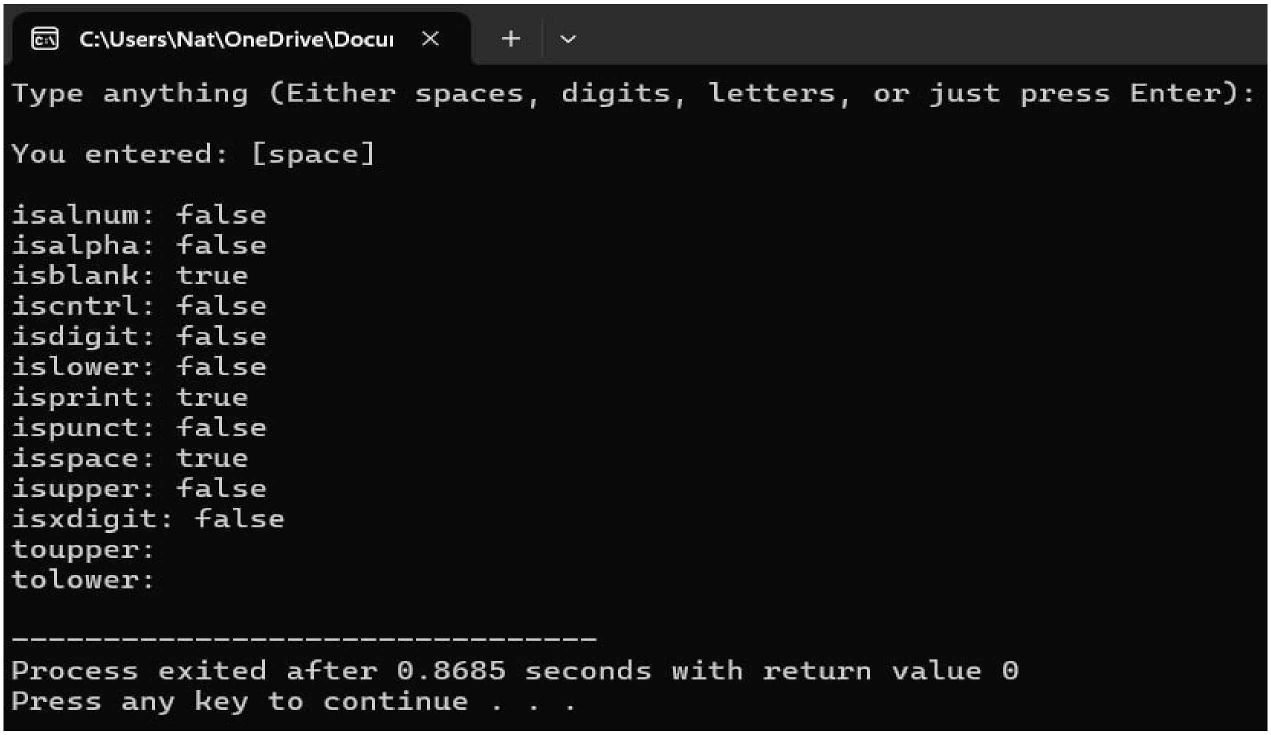
```
Type anything (Either spaces, digits, letters, or just press Enter): N

You entered: N

isalnum: true
isalpha: true
isblank: false
iscntrl: false
isdigit: false
islower: false
isprint: true
ispunct: false
isspace: false
isupper: true
isxdigit: false
toupper: N
tolower: n

--------------------------------
Process exited after 18.84 seconds with return value 0
Press any key to continue . . .
```

Hexadecimal



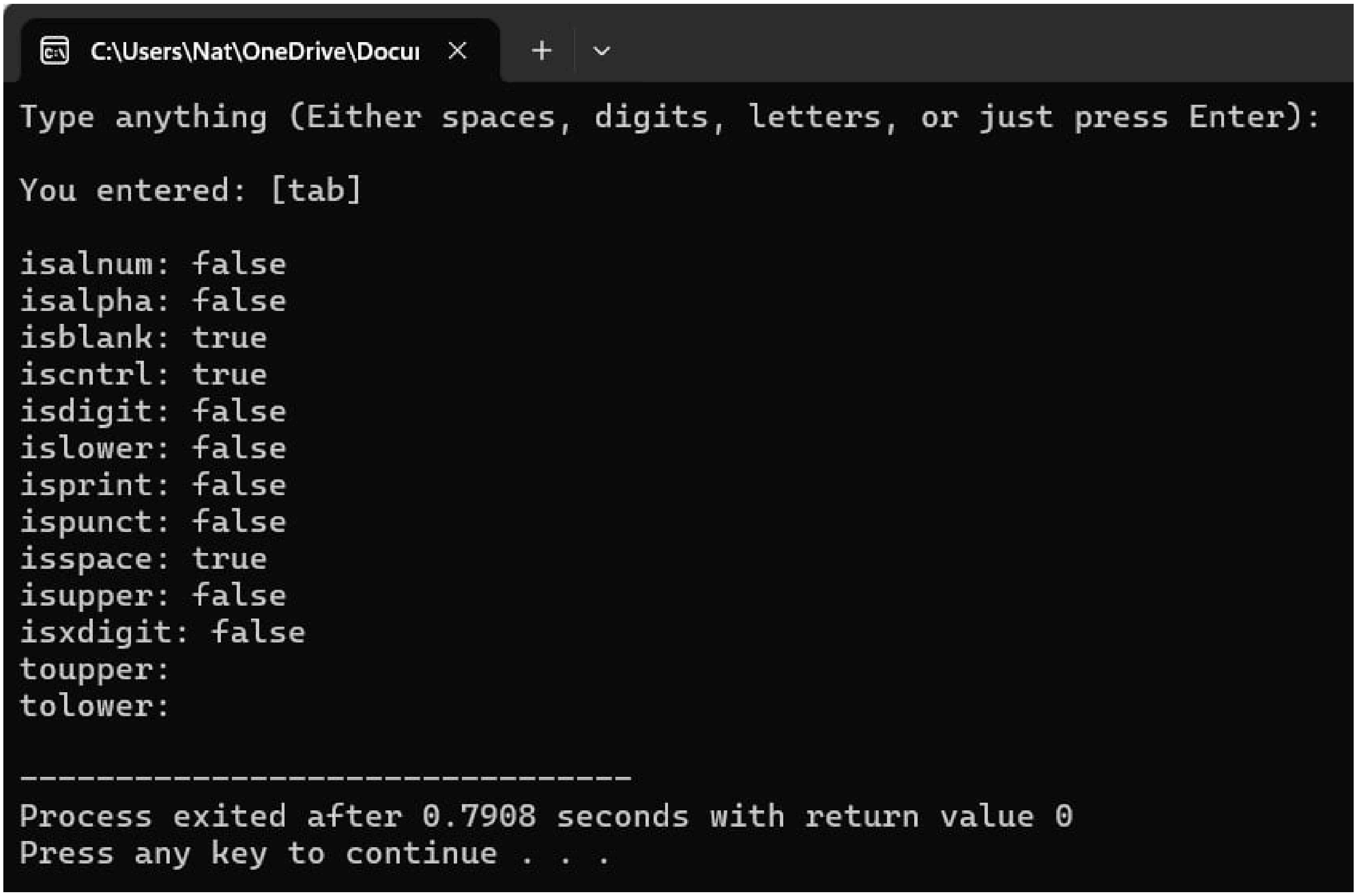
```
Type anything (Either spaces, digits, letters, or just press Enter): F

You entered: F

isalnum: true
isalpha: true
isblank: false
iscntrl: false
isdigit: false
islower: false
isprint: true
ispunct: false
isspace: false
isupper: true
isxdigit: true
toupper: F
tolower: f

--------------------------------
Process exited after 0.8541 seconds with return value 0
Press any key to continue . . .
```

Space



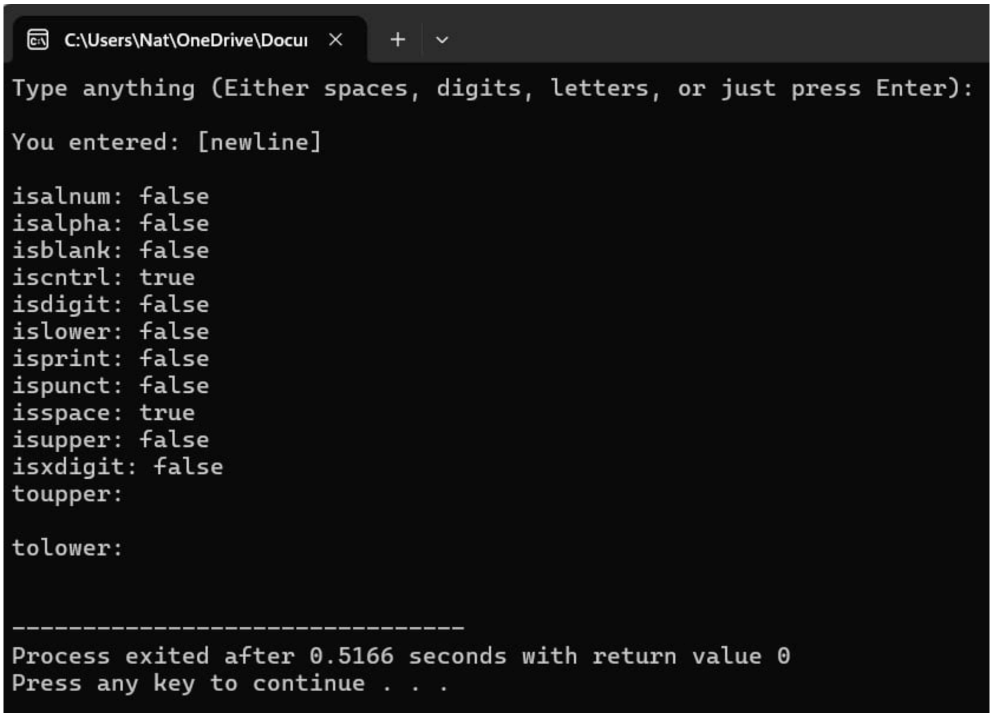
```
Type anything (Either spaces, digits, letters, or just press Enter):

You entered: [space]

isalnum: false
isalpha: false
isblank: true
iscntrl: false
isdigit: false
islower: false
isprint: true
ispunct: false
isspace: true
isupper: false
isxdigit: false
toupper:
tolower:

--------------------------------
Process exited after 0.8685 seconds with return value 0
Press any key to continue . . .
```

Tab



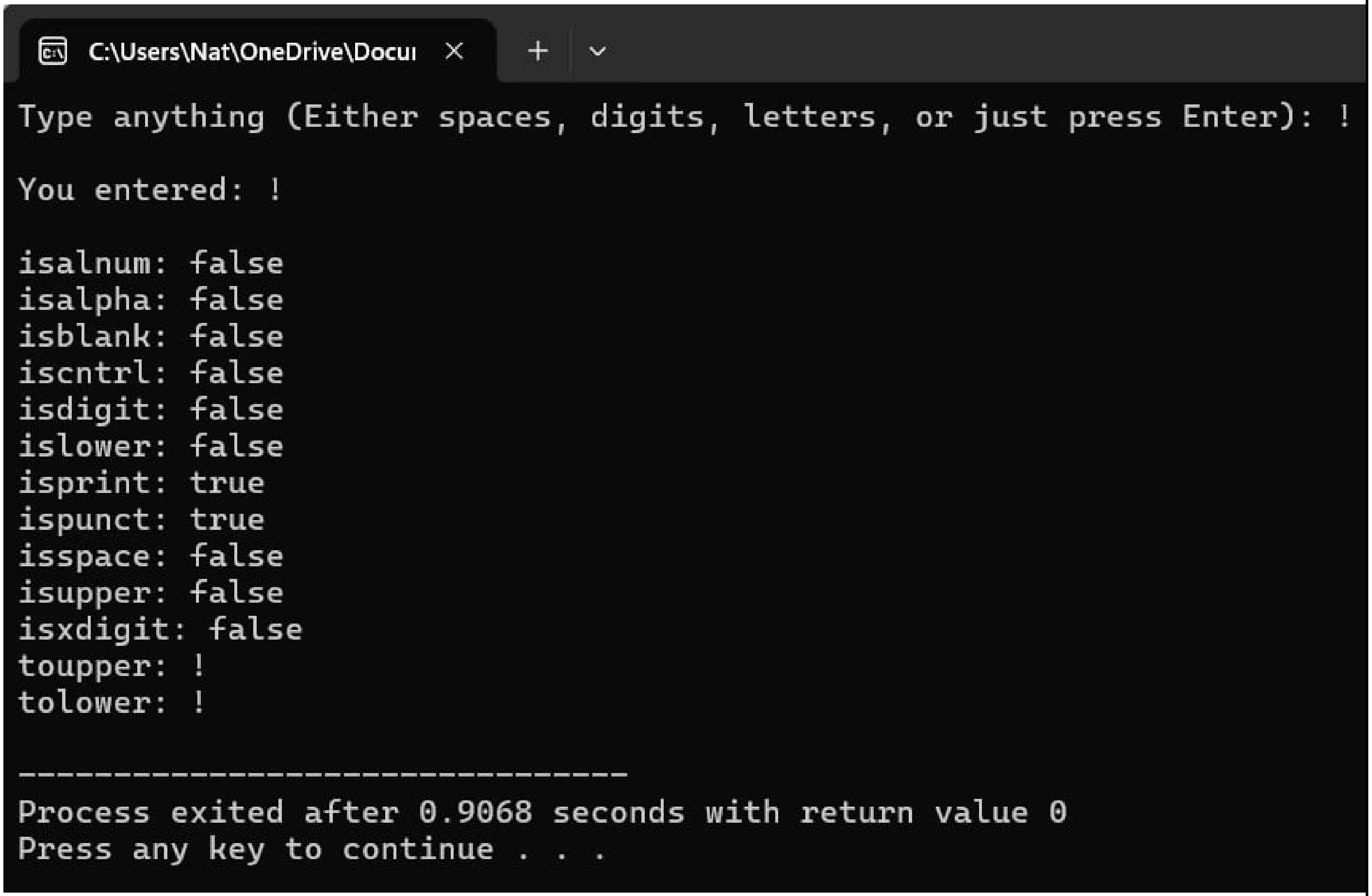
```
Type anything (Either spaces, digits, letters, or just press Enter):

You entered: [tab]

isalnum: false
isalpha: false
isblank: true
iscntrl: true
isdigit: false
islower: false
isprint: false
ispunct: false
isspace: true
isupper: false
isxdigit: false
toupper:
tolower:

--------------------------------
Process exited after 0.7908 seconds with return value 0
Press any key to continue . . .
```

Enter / Newline



```
Type anything (Either spaces, digits, letters, or just press Enter):

You entered: [newline]

isalnum: false
isalpha: false
isblank: false
iscntrl: true
isdigit: false
islower: false
isprint: false
ispunct: false
isspace: true
isupper: false
isxdigit: false
toupper:

tolower:


--------------------------------
Process exited after 0.5166 seconds with return value 0
Press any key to continue . . .
```

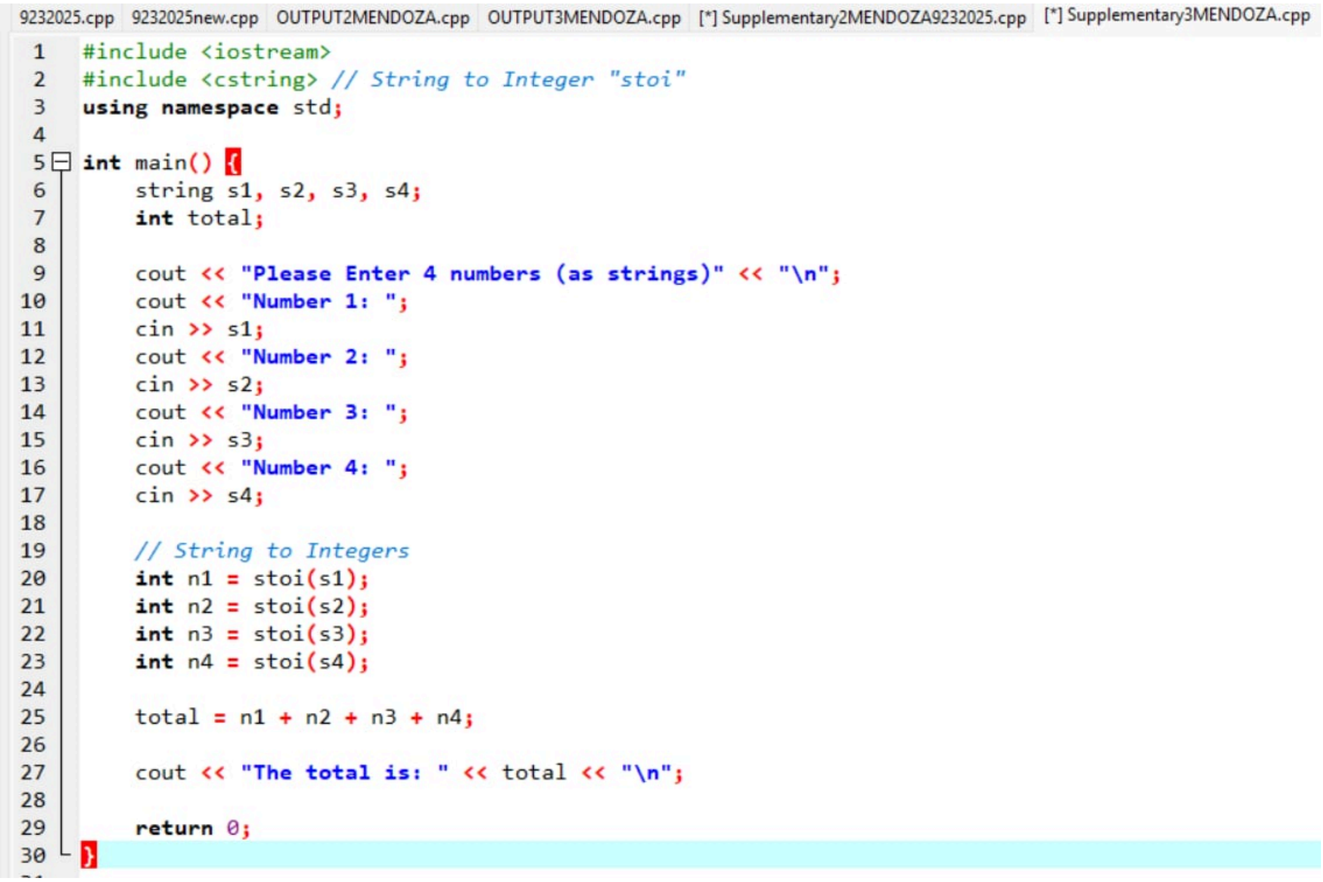
Special Character



```
Type anything (Either spaces, digits, letters, or just press Enter): !

You entered: !

isalnum: false
isalpha: false
isblank: false
iscntrl: false
isdigit: false
islower: false
isprint: true
ispunct: true
isspace: false
isupper: false
isxdigit: false
toupper: !
tolower: !

--------------------------------
Process exited after 0.9068 seconds with return value 0
Press any key to continue . . .
```

3.

Code :

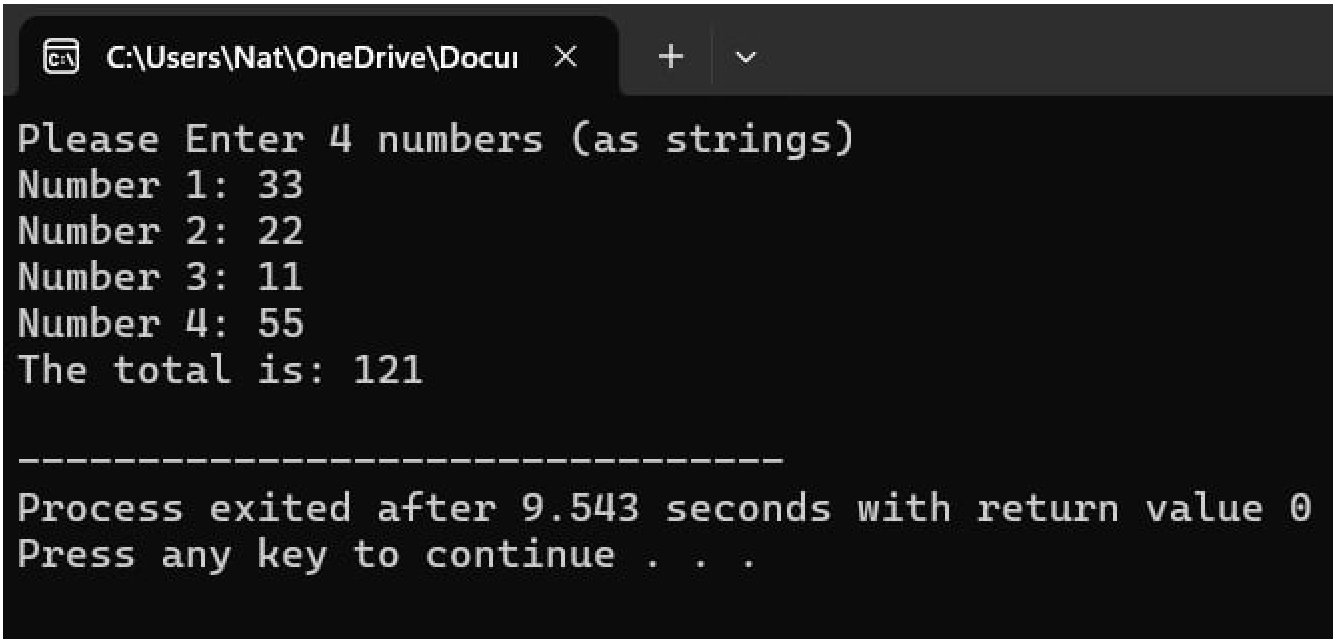

9232025.cpp
9232025new.cpp
OUTPUT2MENDOZA.cpp
OUTPUT3MENDOZA.cpp
[*] Supplementary2MENDOZA9232025.cpp
[*] Supplementary3MENDOZA.cpp

```cpp
#include <iostream>
#include <cstring> // String to Integer "stoi"
using namespace std;

int main() {
    string s1, s2, s3, s4;
    int total;

    cout << "Please Enter 4 numbers (as strings)" << "\n";
    cout << "Number 1: ";
    cin >> s1;
    cout << "Number 2: ";
    cin >> s2;
    cout << "Number 3: ";
    cin >> s3;
    cout << "Number 4: ";
    cin >> s4;

    // String to Integers
    int n1 = stoi(s1);
    int n2 = stoi(s2);
    int n3 = stoi(s3);
    int n4 = stoi(s4);

    total = n1 + n2 + n3 + n4;

    cout << "The total is: " << total << "\n";

    return 0;
}
```

Code Analysis :

So my program begins by including the header file <iostream> which allows the use of input and output operations such as cin and cout. It also includes <cstring>. The using namespace std; line is used so we don’t need to prefix std:: every time we use standard library functions like cin, cout, or String to Integer (stoi). Then inside the main() function, four string variables s1, s2, s3, and s4 are declared, along with an integer variable total that will hold the sum of the numbers. The program then prompts the user to enter four numbers, one at a time, treating them as strings. Each input is stored in the corresponding string variable. The use of strings here shows that the program accepts input not immediately as numbers, but as text. Next, the program converts these string inputs into integers using the function stoi, which stands for string to integer. The strings s1, s2, s3, and s4 are each converted into integer variables n1, n2, n3, and n4. This step is important because mathematical operations like addition cannot be performed directly on string data types, only on numeric ones. After conversion, the program adds the four integers together and stores the result in the variable total. Finally, it displays the computed total using cout, providing the user with the sum of the four entered numbers. The program then ends successfully with return 0;.

Code Output :



```
Please Enter 4 numbers (as strings)
Number 1: 33
Number 2: 22
Number 3: 11
Number 4: 55
The total is: 121

--------------------------------
Process exited after 9.543 seconds with return value 0
Press any key to continue . . .
```

8. Conclusion

In this Hands-on activity I was able to successfully apply the functions of each character by using <cctype> libraries and <cstrings>. Because I was able to utilize it with the input and output functions by using such libraries and being able to do conversion and processing. Additionally, I gained new knowledge such as that C++ Languages can handle characters and strings at the same time such as converting strings to main from my supplementary activities. Also, it is interesting and challenging to try 13 types of functions and implement it into 1 code itself. This would help me as a Computer Engineering student to know more special characters, digits, and etc. In which this lesson will determine which and what kind of character was implemented as it consists of 13 functions just by telling as true or false. In addition, I learned how to accept string inputs and convert them into integers using (stoi) or String to Integer so that mathematical operations like addition can be performed. This showed me the difference between handling input as text and as numbers. Using loops, I was also able to repeat checks multiple times without rewriting the same code, which makes programs more efficient and easier to manage. Overall, these activities helped me realize that handling input carefully whether characters or strings and knowing how to convert or process them properly is very important in programming. It also made me appreciate how useful C++ libraries like <cctype> and <string> are for performing common tasks in a simple and reliable way.