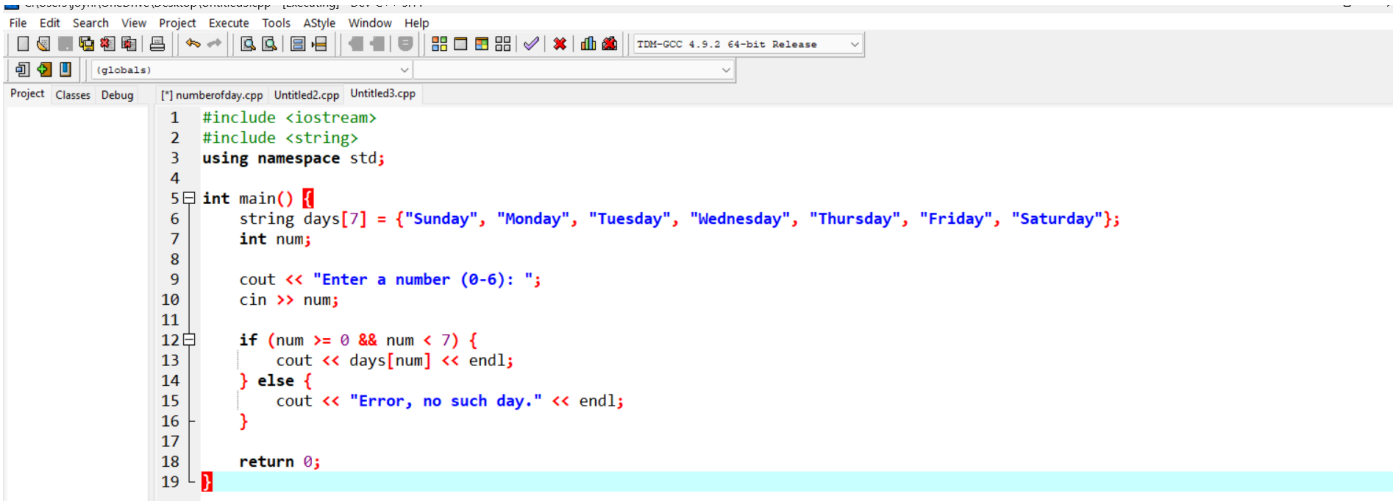
Activity No. n 4.3	
Replace with Title	
Course Code: CPE11S1	**Program:** Computer Engineering
Course Title: Programming Logic and Design	**Date Performed: 09/16/25**
Section: CPE11S1	**Date Submitted:09/16/25**
Name(s): Candy M. Nieles	**Instructor: Engr. Jimlord M. Quejado**

6. Output

7. Supplementary Activity

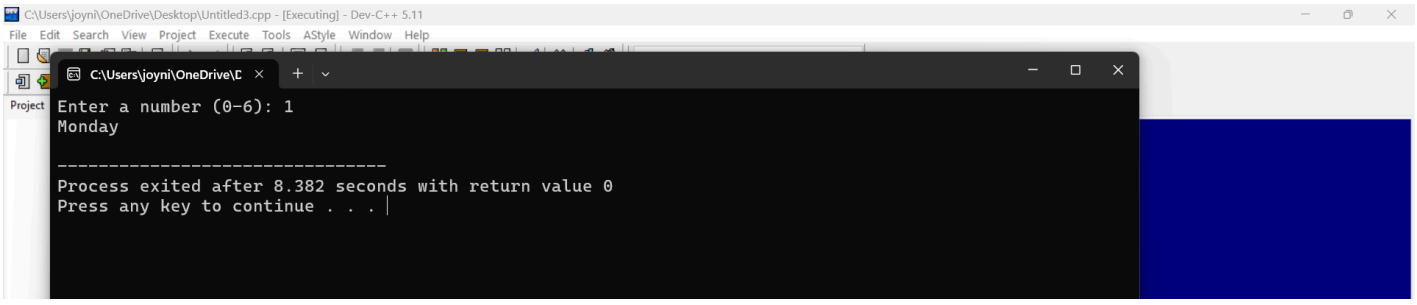
1. Write a program that asks for a number from the user and prints which day of the week that number corresponds to. The days are indexed from 0 (Sunday) to 6 (Saturday). Before the program gets a value from the array, it must first check if the given day is greater than or equal to zero and less than 7. If not, it should print the message: "Error, no such day." Your version of the program must print the same result as the expected output.

INPUT:



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

int main() {
    string days[7] = {"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"};
    int num;

    cout << "Enter a number (0-6): ";
    cin >> num;

    if (num >= 0 && num < 7) {
        cout << days[num] << endl;
    } else {
        cout << "Error, no such day." << endl;
    }

    return 0;
}
```

OUTPUT:



```
Enter a number (0-6): 1
Monday

--------------------------------
Process exited after 8.382 seconds with return value 0
Press any key to continue . . .
```

EXPLATION:

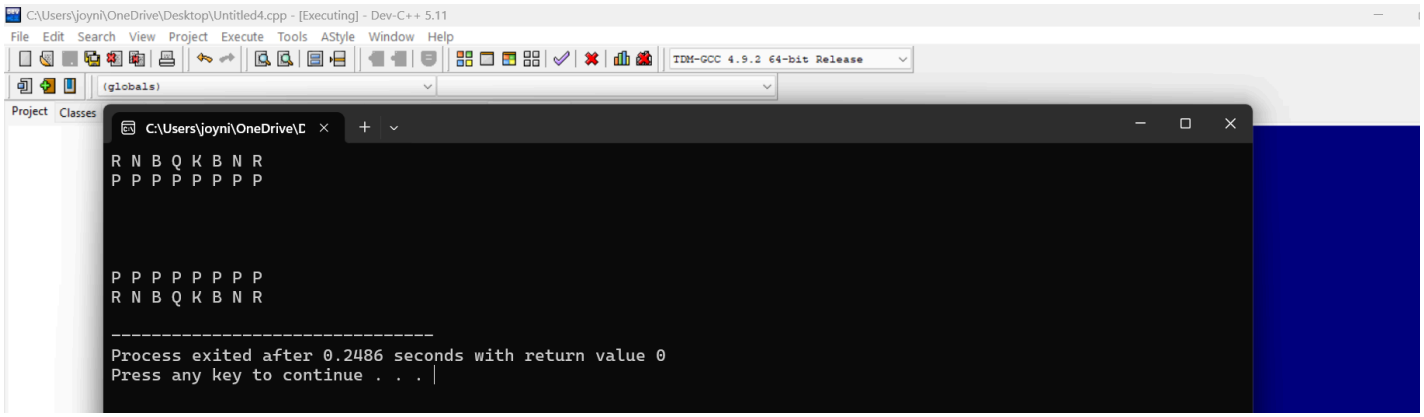
In this program, we use an array to store the names of the days and if-else statements to handle user input. #include, which provides a function to input/output of program #include. It allows you to use the std::string data type for handling the text. Then Array Declaration and initializes the array name of numberOfDays. Int main () { Where the main function where the program is going to begin declares an array name that can hold 7 string elements cout << “Enter a number (0-6):“ to the console, prompting the user to input cin>>num; the value entered by the user from the console and stored in the num variable. The next step is if (num >= 0 && num<7) means this ensures that the entered number is a valid index for the day's array, which has indices from 0 to 6 (num >=0 and num <7 must be true for the if block to execute. Then put cout<<” Error no such day.”<

2. Write a program that creates a chessboard, sets all the pieces on it and then displays the contents of the board. Create a two-dimensional array, fill it with data and print a letter when a piece is on the field and a space when there is no piece. Store one letter for one piece. For now, we don't need any information about the color of the pieces. The starting positions (with letters which symbolize each piece) for all pieces are: The rooks (R) are placed on the outside corners, right and left edge (white on the 1st and black on the 8th line). The knights (N) are placed immediately inside of the rooks. The bishops (B) are placed immediately inside of the knights. The queen (Q) is placed on the central square of the same color as that of the player: white queen on the white square and black queen on the black square. Both stand on the d rank: white queen on the d1 field and black queen on the d8 field. The king (K) takes the vacant spot next to the queen. The pawns (P - not the official symbol, but you need to print something) are placed one square in front of all of the other pieces. Your version of the program must print the same result as the expected output.

INPUT:

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

int main() {
    // 8x8 chessboard filled with spaces
    char board[8][8];

    // Fill with spaces
    for (int i = 0; i < 8; i++) {
        for (int j = 0; j < 8; j++) {
            board[i][j] = ' ';
        }
    }

    // Black pieces (top side)
    char backRow[8] = {'R', 'N', 'B', 'Q', 'K', 'B', 'N', 'R'};
    for (int j = 0; j < 8; j++) {
        board[0][j] = backRow[j]; // first row
        board[1][j] = 'P';        // pawns
    }

    // White pieces (bottom side)
    for (int j = 0; j < 8; j++) {
        board[7][j] = backRow[j]; // last row
        board[6][j] = 'P';        // pawns
    }

    // Print board
    for (int i = 0; i < 8; i++) {
        for (int j = 0; j < 8; j++) {
            cout << board[i][j] << " ";
        }
        cout << endl;
    }

    return 0;
}
```

OUTPUT:



EXPLANATION:

#include <iostream> using namespace std; it includes the input/output of the program and avoid the need to prefix standard library element. Int main() { The main function were the program going to begins declares a 8x8 2D which represent in chessboard. The loop iterates though each cell of the arrays and sets it to a space character “ means the board to be empty and sets the first row to the corresponding piece from the backRow array and the second raw is representing the black pawns. The purpose of this loop is to position the white pieces on the board's underside. But there's a mistake: it makes use of the backRow array, which holds the black pieces. Additionally, the last row should be assigned the reversed backRow. board[7][j] = backRow[j]; : This line puts the black pieces in the last row (board[7]) in the wrong

way. The white pieces should be placed there, but they are not in any array.
board[6][j] = 'P';: This line puts black pawns in the second-to-last row (board[6]) in the wrong way. White pawns should be placed there.
Lines 29–34: Printing the Board. Then Return 0; means your input is done and run it to see the output results.

8. Conclusion

Based on the concepts of sorting and searching arrays, we can conclude that these two operations are fundamentally interconnected and essential for efficient data management. Sorting an array arranges its elements into a specific order, such as numerical or alphabetical, which is a critical preprocessing step. While sorting itself requires computational time, it enables significantly faster subsequent operations.Elements of an array can be accessed by their position (called index) in the sequence. In C++, indexes of an array starts from 0 instead of 1.Stated differently, arrays in C++ language are defined at declaration. The type name, variable name, and size (in brackets) are typically provided when declaring an array.