Activity 4.3	
Sorting and Searching Arrays	
Course Code: CPE 007	**Program:** Computer Engineering
Course Title: Programming Logic and Design	**Date Performed: 9/16/2025**
Section: CPE11S1	**Date Submitted: 9/18/2025**
Name(s): Juan Paulo C. Lara	**Instructor: Engr. Jimlord M. Quejado**

6. Output

7. Supplementary Activity

Problem 1

Code

```
1  #include <iostream>
2  using namespace std;
3  int main() {
4      string dayName[7] = {"Sunday", "Monday", "Tuesday",
5      "Wednesday", "Thursday", "Friday", "Saturday"};
6
7      int dayNumber;
8      cout << "Input day number (0-6): ";
9      cin >> dayNumber;
10
11     if (dayNumber >= 0 && dayNumber < 7) {
12         cout << "Output day name: " << dayName[dayNumber] << endl;
13     } else {
14         cout << "Error, no such day." << endl;
15     }
16
17     return 0;
18 }
```

Output

```
Input day number (0-6): 0
Output day name: Sunday
```

```
Input day number (0-6): 5
Output day name: Friday
```

```
Input day number (0-6): 12
Error, no such day.

--------------------------------
Process exited after 0.4795 seconds with return value 0
Press any key to continue . . .
```

Analysis

When running this program, line 4 initializes a 2D array containing the names of the days of the week. Line 7 declares int as dayNumber to act as an index listing for the names given by the array above. Line 8 outputs a message telling the user to input a number of days from 0 (Sunday) to 6 (Saturday) and line 9 is the input line. Line 11 uses an if function to

check if the inputted day is valid (0-6). Line 12 prints out the name of the input day using the array index. Line 13 checks if there is a response outside of the allowed values (e.g. -1 and 7) and tells line 14 to output a message telling the user the input is invalid.

Problem 2

Code

```
1   #include <iostream>
2   using namespace std;
3   #define boardX 8
4   #define boardY 8
5   int main(){
6       char board[boardX][boardY] = {
7           {'R','N','B','Q','K','B','N','R'},
8           {'P','P','P','P','P','P','P','P'},
9           {' ',' ',' ',' ',' ',' ',' ',' '},
10          {' ',' ',' ',' ',' ',' ',' ',' '},
11          {' ',' ',' ',' ',' ',' ',' ',' '},
12          {' ',' ',' ',' ',' ',' ',' ',' '},
13          {'P','P','P','P','P','P','P','P'},
14          {'R','N','B','Q','K','B','N','R'}
15      };
16
17      for (int i = 0; i < 8; i++) {
18          for (int j = 0; j < 8; j++) {
19              cout << board[i][j] << " ";
20          }
21          cout << endl;
22      }
23      return 0;
24  }
```

Output

```
R N B Q K B N R
P P P P P P P P




P P P P P P P P
R N B Q K B N R

--------------------------------
Process exited after 0.1465 seconds with return value 0
Press any key to continue . . .
```

Analysis

Starting with line 3 and 4, the x and y parts of the 2d array board are defined to 8 to form a 8x8 grid at runtime. Line 6 initializes the 2d array board containing the x and y axis. An opening bracket is placed before the rows of arrays containing the characters. Taken from the placement given by the sample output, Line 7 contains the elements for the first row containing the rook, knight, bishop, queen, and king, and the bishop, knight, and rook repeated. Line 8 contains the repeated elements of pawn. Starting from Line 9 to 12, rows of elements containing blank spaces are placed. Line 13 repeats the same contents as line 8 and line 14 repeats the elements of line 7 for the black side. After that, an outer for loop at line 17 scans the board horizontally and an inner for loop at line 18 scans the board vertically. Line 19 outputs the elements with spaces and endl command at line 21 stops and moves over to the next line and continues the loop.

8. Conclusion

Concluding this activity, I learned the ways of placing data from arrays, more specifically the 2D arrays. The features given by 2d arrays are an additional [ ] bracket when declaring a row of elements and another { } curly brackets to allow containing multiple rows of arrays. With this, I can initialize one large array containing all the data to search, manipulate, and output instead of declaring more and in turn, reducing the amount of lines and the time it takes to compile and run. After the class discussions, I also learned a new method of initializing arrays containing names with the use of string commands. Initially, I used a char command then a two dimensional declaration of an array that is just one row of the days of the week. But this time, I replaced the declaring command char with string and removed the second bracket to streamline the code. Overall, learning these concepts, I realized that the 2d array is basically an x and y axis array containing grids of information that allow for other processes like a for loop to read, manipulate, and display the contained data.