Hands-on Activity 4.3	
Sorting and Searching Arrays	
Course Code: CPE 007	**Program:** Computer Engineering
Course Title: Programming Logic and Design	**Date Performed: 09/16/2025**
Section: CPE11S1	**Date Submitted: 09/17/2025**
Name(s): Canoy Hail B.	**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.

Code :

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

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

    cout << "Enter Number (0-6) : ";
    cin >> number_of_days;

    if ( number_of_days > -1 && number_of_days < 7) {
        cout << days[number_of_days] << endl;

    }
        else {
            cout << "Error, No such day.";
        }




    return 0;
}
```

Output :

```
Enter Number (0-6) : 0

-----Monday-----

--------------------------------
Process exited after 2.636 seconds with return value 0
Press any key to continue . . .
```

```
Enter Number (0-6) : 7
Error, No such day.
--------------------------------
Process exited after 4.114 seconds with return value 0
Press any key to continue . . .
```

Explanation :

- First I entered **int number_of_days**; this will store all the number of choices you can make. Second I entered all possible value of the choices in the array string which is **string days[7] = {"\n-----Monday-----", "\n-----Tuesday-----", "\n-----Wednesday-----", "\n-----Thursday-----", " \n-----Friday-----", "\n-----Saturday-----", "\n-----Sunday-----"};** , Third is **cout << "Enter Number (0-6) : ";** this is for the output to print and have an direction for the user to do. **if (number_of_days > -1 && number_of_days < 7) {** this is for the program to see if the user will enter a value between 0 to 6 and then will print the exact array choice.
cout << days[number_of_days] << endl; } so the output will print the value of days following the number_of_days. Then the last step is **else { cout << Error, No such day." ;** to print Error, No such day everytime a user enters a value greater than 7 or less than 0.

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.

Code :

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

int main() {
    const int size = 8;
    char board[size][size];

    for (int r = 0; r < size; r++) {
        for (int c = 0; c < size; c++) {
            board[r][c] = ' ';
        }
    }

    board[0][0] = board[0][7] = 'R';
    board[7][0] = board[7][7] = 'R';

    board[0][1] = board[0][6] = 'N';
    board[7][1] = board[7][6] = 'N';

    board[0][2] = board[0][5] = 'B';
    board[7][2] = board[7][5] = 'B';

    board[0][3] = 'Q';
    board[7][3] = 'Q';

    board[0][4] = 'K';
    board[7][4] = 'K';

    for (int i = 0; i < size; i++) {
        board[1][i] = 'P';
        board[6][i] = 'P';
    }

    for (int r = 0; r < size; r++) {
        for (int c = 0; c < size; c++) {
            cout << board[r][c] << " ";
        }
        cout << endl;
    }

    return 0;
}
```

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 1.037 seconds with return value 0
Press any key to continue . . .
```

Explanation :

- First, we are gonna change the chessboard size using const int size to 8 then we create an array using char to hold letters. then we are going to do the first loop which is **for (int r = 0; r < size; r++) { for (int c = 0; c < size; c++) {board[r][c] = ' ';** for it to put spaces and columns that we need in the chess board. After that we are going to add the rooks using **board[0][0] = board[0][7] = 'R'; board[7][0] = board[7][7] = 'R'** to place the rooks then repeat it but change the values **board[0][1] = board[0][6] = 'N'; board[7][1] = board[7][6] = 'N';** for the Knights, **board[0][2] = board[0][5] = 'B'; board[7][2] = board[7][5] = 'B';** for the bishops, **board[0][3] = 'Q'; board[7][3] = 'Q';** for the queen, **board[0][4] = 'K'; board[7][4] = 'K';** for the king, and for the pawn we are going to do another loop to**for (int i = 0; i < size; i++) { board[1][i] = 'P';board[6][i] = 'P'; }** and finally we are going to do the last loop to print everything in r 0-7 and c 0-7 and to do that we are going to put **for (int r = 0; r < size; r++) { for (int c = 0; c < size; c++) {cout << board[r][c] << " ";}cout << endl; }.**

8. Conclusion

- In this Hands-on Activity 4.3 I've learned a lot more about arrays with choices, how to loop better, and how to make a chessboard-like output by understanding the columns and row. In the first supplementary activity I learned to use a condition in the array for the day and number of days to match, While in the second activity I learned more about columns and rows like board [ ] [ ], and how to properly use them to structure a chess board. These activities helped me practice loops and arrays, and showed me how to organize characters in rows and columns. I understand it, but still not that much. While I was creating the code I really always found it hard to do it from scratch. I need to memorize the structures more so I can make my code simpler and easier to understand so I won't make that much of a mistake.