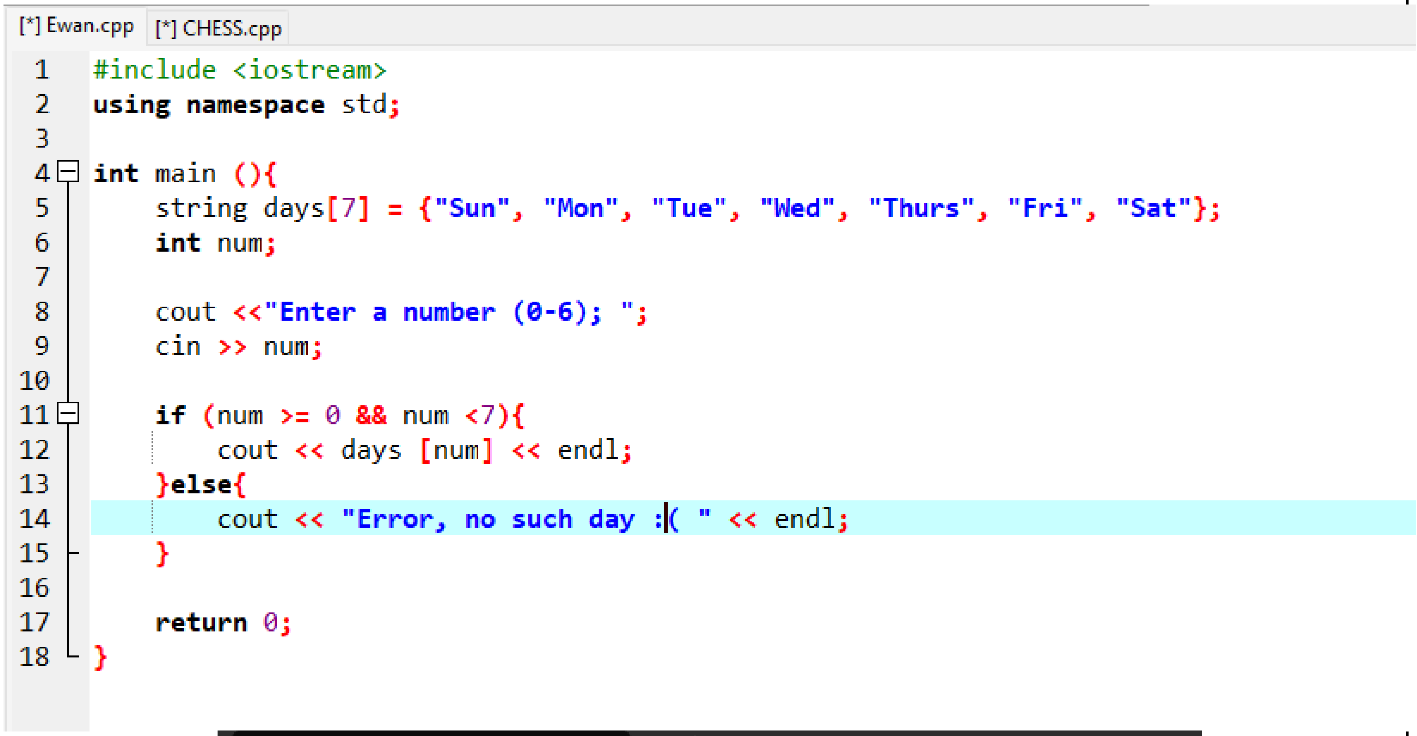
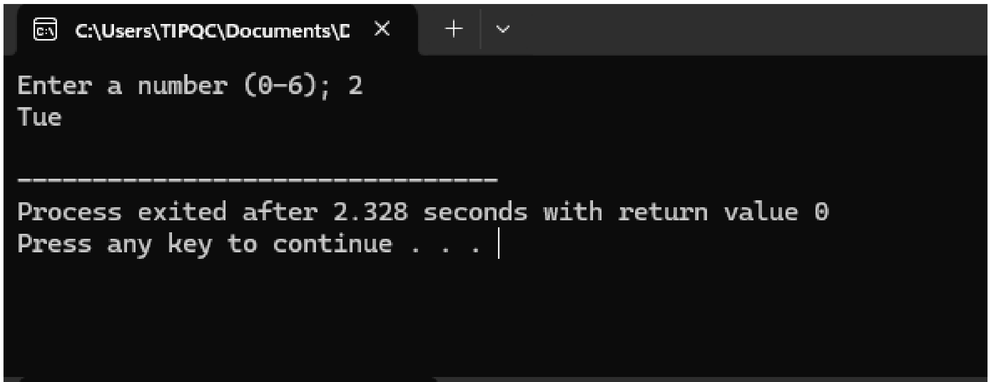
Hands on Activity No. 4.3	
Sorting and Searching Arrays	
Course Code: CPE007	Program: Computer Engineering
Course Title: Programming Logic and Design	Date Performed: Sep. 16, 2025
Section: CPE11S1	Date Submitted: Sep. 18, 2025
Name(s): Tobias, Lawrence C.	Instructor: Engr. Jimlord M. Quejado
6. Output	

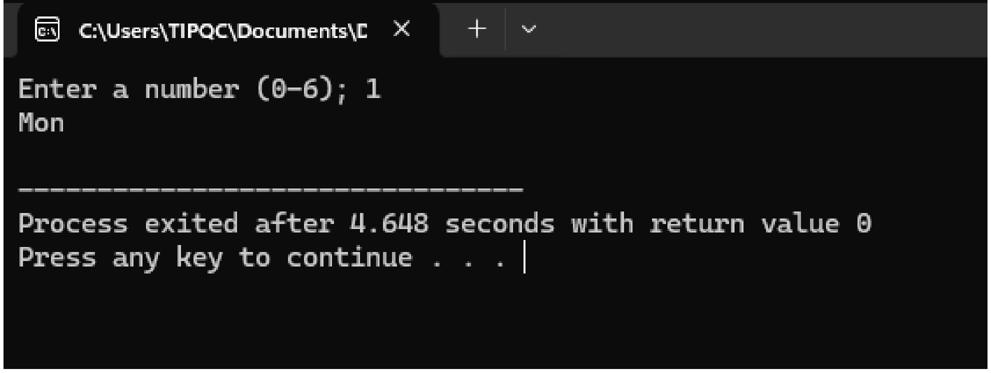
1.



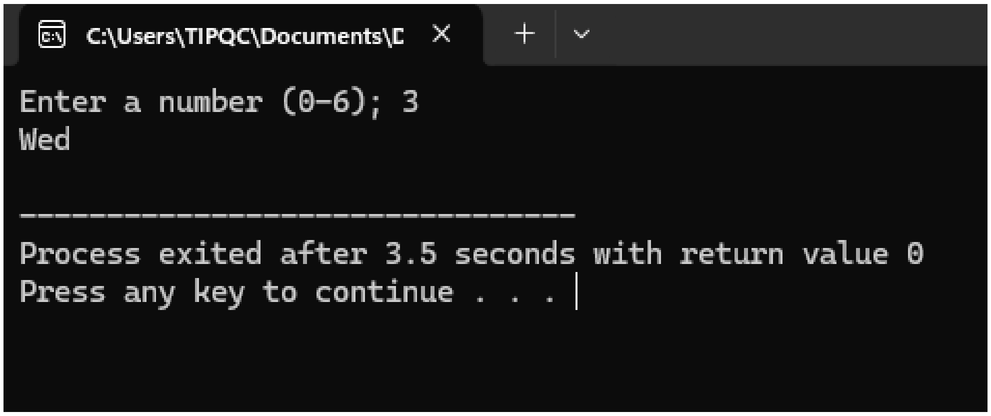
```cpp
#include <iostream>
using namespace std;

int main (){
    string days[7] = {"Sun", "Mon", "Tue", "Wed", "Thurs", "Fri", "Sat"};
    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;
}
```



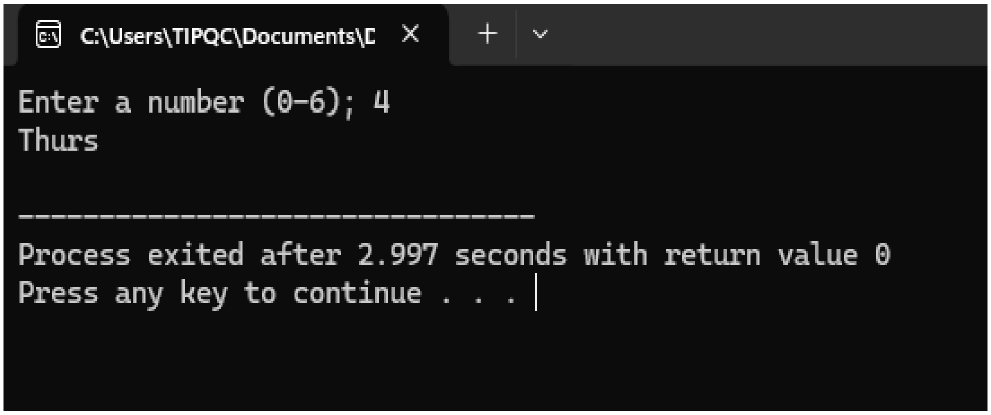
```
Enter a number (0-6); 2
Tue

--------------------------------
Process exited after 2.328 seconds with return value 0
Press any key to continue . . .
```



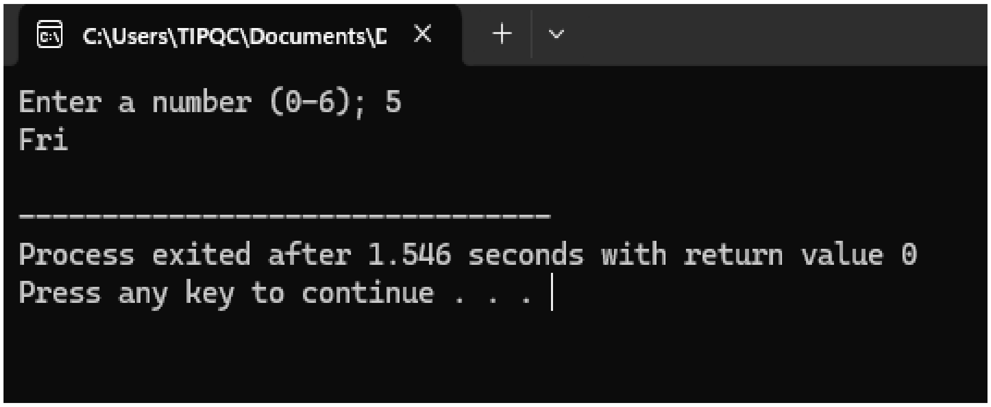
```
Enter a number (0-6); 1
Mon

--------------------------------
Process exited after 4.648 seconds with return value 0
Press any key to continue . . .
```



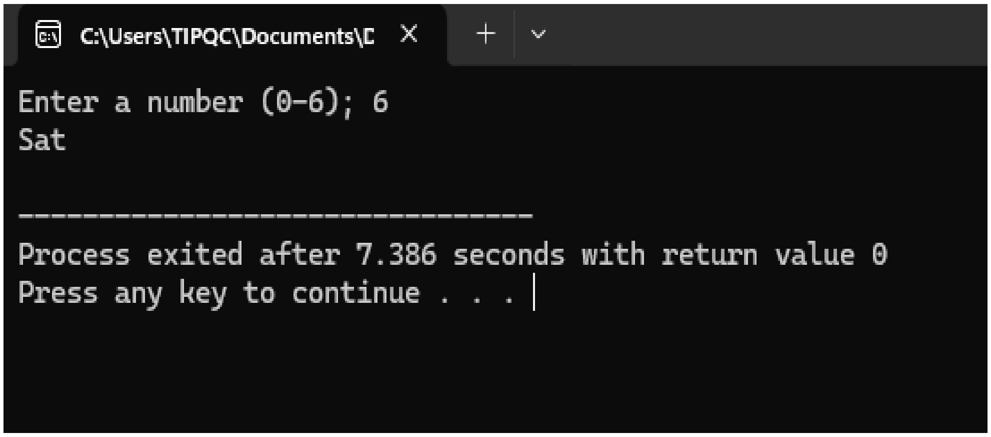
```
Enter a number (0-6); 3
Wed

--------------------------------
Process exited after 3.5 seconds with return value 0
Press any key to continue . . .
```



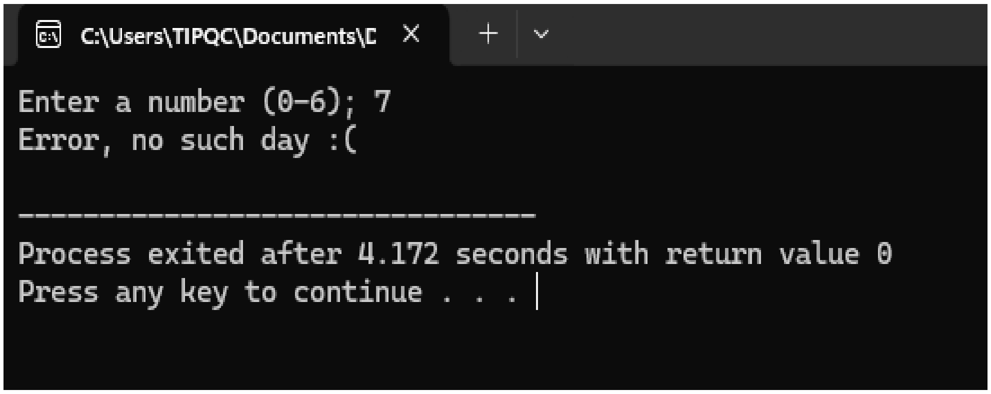
```
Enter a number (0-6); 4
Thurs

--------------------------------
Process exited after 2.997 seconds with return value 0
Press any key to continue . . .
```



```
Enter a number (0-6); 5
Fri

--------------------------------
Process exited after 1.546 seconds with return value 0
Press any key to continue . . .
```

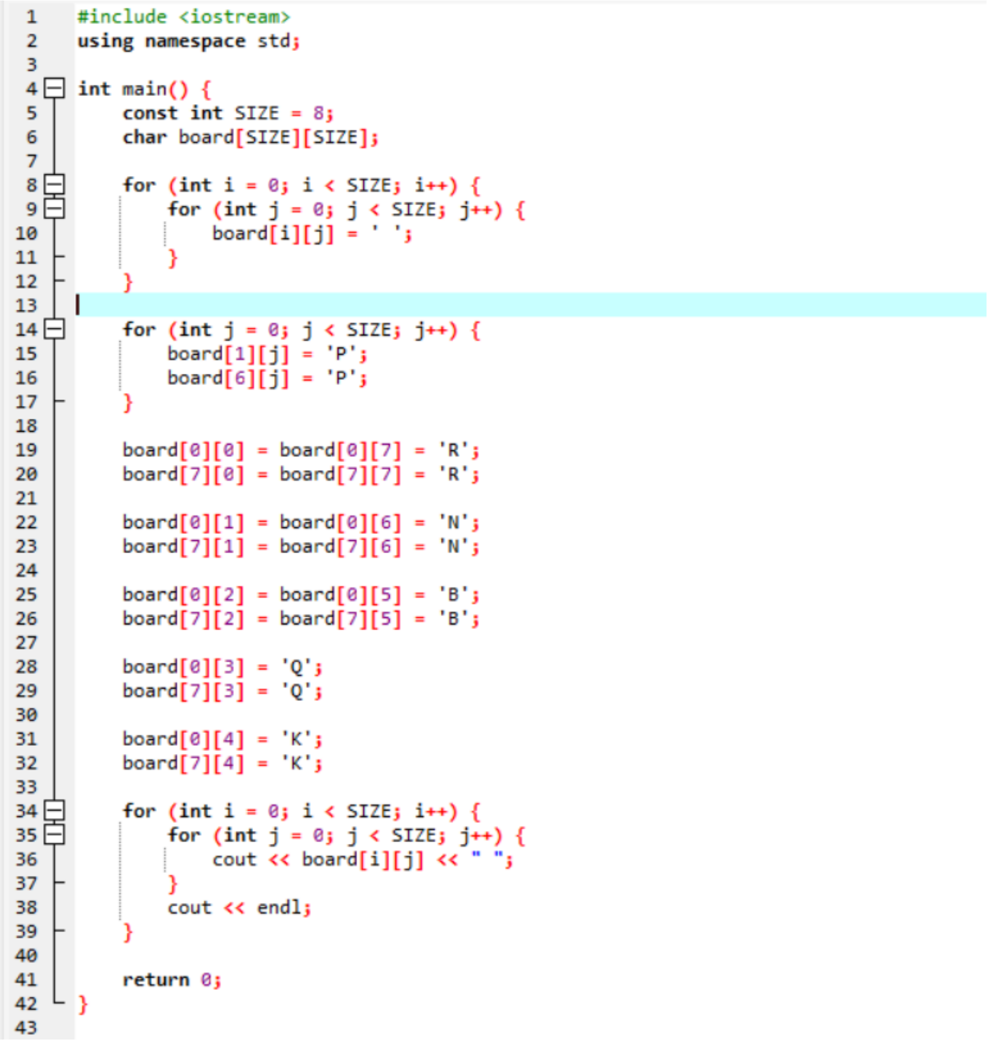


```
Enter a number (0-6); 6
Sat

--------------------------------
Process exited after 7.386 seconds with return value 0
Press any key to continue . . .
```

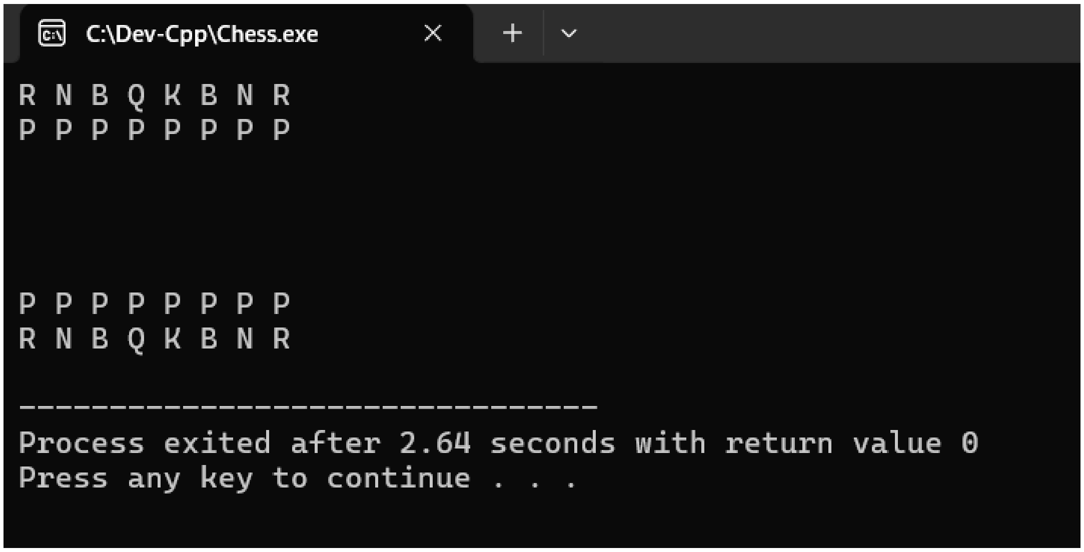


```
Enter a number (0-6); 7
Error, no such day :(

--------------------------------
Process exited after 4.172 seconds with return value 0
Press any key to continue . . .
```

2.



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

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

    for (int i = 0; i < SIZE; i++) {
        for (int j = 0; j < SIZE; j++) {
            board[i][j] = ' ';
        }
    }

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

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

    return 0;
}

```



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

7. Supplementary Activity

1. The program asks the user to provide a number so it can identify what day of the week it is. The number can be between 0 and 6 (0 = Sunday, 1 = Monday, etc.) and will produce an error message if outside these numbers. The program uses a list that has the seven days of the week represented, then asks the user for a number. The program uses an if statement to get user input and check if it's a valid number; that means the condition makes sure the number is not negative and is lower than 7, so the range of numbers is between 0 and 6. If the number is not between 0 and 6, the else part of the code will run and produce "Error, no such day." In other words, the program will run by producing the correct day of the week if the number is correct or an error if invalid.

2. The program creates a chessboard that is formatted as a 2D array with a size of 8x8, using a letter to represent each chess piece. First, I created a constant "size" and set it equal to the value of 8 as there are 8 rows and 8 columns on a chessboard. Then, I created a 2D array named board to symbolize the chessboard, and I used nested for loops to fill it with spaces so that all squares begin empty. After that, I filled the first two rows with black pieces using lowercase letters and the last two rows with white pieces using uppercase letters. I also filled the second to last row with white pawns. Lastly, I utilized another for loop with if-else statements to print the board. Ultimately, the program displayed the chessboard accurately with all the pieces in their original positions.

8. Conclusion

From this activity, I learned how to use arrays in different ways. In the first program, I used a one-dimensional array to store the days of the week. The program asked for a number and showed the right day if it was between 0 and 6, or an error if not. In the second program, I used a two-dimensional array to make a chessboard. I filled it with spaces first, then placed the black pieces on top and the white pieces on the bottom, with pawns in their rows. I used loops to print everything so it looked like a real chessboard. I struggled a bit with the chessboard since it was new to me, but I was glad I made it work. Overall, I learned a lot from both tasks and I know I can still get better at making cleaner codes.