Certainly! Here's a line-by-line explanation of the code:

cpp

Copy code

#include <bits/stdc++.h>

This line includes a standard C++ library header, which provides access to various standard libraries commonly used in C++ programs.

cpp

Copy code

using namespace std;

This line specifies that the code is using the std namespace, so you don't need to prepend standard library elements with std::.

cpp

Copy code

struct MinMax

{

  int minimum;

  int maximum;

};

This block defines a structure named MinMax with two integer members: minimum and maximum. It will be used to store the minimum and maximum values found in an array.

cpp

Copy code

MinMax minimumMaximum(int arr[], int left, int right)

{

  // Function to find the minimum and maximum in a given array within a specific range.

}

This line declares a function named minimumMaximum that takes an integer array arr, and two integers left and right to specify the range in the array. It returns a MinMax structure with the minimum and maximum values found within the specified range.

cpp

Copy code

```
if (left == right) /// single element
{
   return {arr[left], arr[right]};
}
```

This code checks if left and right are equal, which means there is only one element in the specified range. In this case, it returns a MinMax structure with both minimum and maximum set to the value at index left.

cpp

Copy code

```
else if (left + 1 == right) /// two elements
{
   // Check if there are two elements in the range.
   if (arr[left] < arr[right])
      return {arr[left], arr[right]};
   else
      return {arr[right], arr[left]};
}
```

In this section, the code checks if there are exactly two elements in the range. If so, it compares the values at indices left and right, and returns a MinMax structure with the smaller value as minimum and the larger value as maximum.

cpp

Copy code

```
else /// more than two elements
{
   // When there are more than two elements in the range.
   int mid = (left + right) / 2;
   MinMax leftSide = minimumMaximum(arr, left, mid);
   MinMax rightSide = minimumMaximum(arr, mid + 1, right);
```

```
    MinMax finalResult;
    // ...
}
```

If neither of the above conditions is met, it means there are more than two elements in the range. The code calculates the middle index mid and then recursively calls the minimumMaximum function on the left and right halves of the array. The results are stored in leftSide and rightSide. A MinMax structure named finalResult is declared to store the final minimum and maximum values found within the entire range.

cpp

Copy code

```
if (leftSide.minimum < rightSide.minimum)
    finalResult.minimum = leftSide.minimum;
else
    finalResult.minimum = rightSide.minimum;
```

Here, the code determines the minimum value by comparing the minimum values from the left and right sides. The smaller of the two values is assigned to finalResult.minimum.

cpp

Copy code

```
if (leftSide.maximum > rightSide.maximum)
    finalResult.maximum = leftSide.maximum;
else
    finalResult.maximum = rightSide.maximum;
```

This section is similar to the previous one but determines the maximum value by comparing the maximum values from the left and right sides. The larger of the two values is assigned to finalResult.maximum.

cpp

Copy code

```
return finalResult;
```

Finally, the minimumMaximum function returns the finalResult, which contains the minimum and maximum values for the entire range.

cpp

Copy code

```
int main()
{
    int arr[10] = {6, 9, 8, 4, 5, 1, 2, 3, 4, 5};
    int n = 10;
    MinMax result = minimumMaximum(arr, 0, n - 1);
    cout << result.minimum << " " << result.maximum << endl;
    return 0;
}
```

In the main function, an integer array arr is defined and initialized with values. The variable n is set to 10, representing the length of the array. The minimumMaximum function is called with the array, starting index 0, and ending index n - 1 (the full array). The result is stored in a MinMax structure called result. Finally, the code prints the minimum and maximum values found by the minimumMaximum function using cout.