Explanation of the Insertion Sort file:

The given file demonstrates the implementation of the Insertion Sort algorithm to sort an array of integers in ascending order. The `insertionSort` function is defined to perform the Insertion Sort algorithm. This function takes two parameters: an integer array `arr` and its size `size`. Inside the `insertionSort` function, a `for` loop is used to iterate over the array elements starting from the second element (`i = 1`) since the first element is already considered sorted.The variable `key` is used to store the current element to be inserted at its correct sorted position in the array. The variable `j` is initialized to the index before the current element (`i - 1`) to compare and shift elements to the right. A `while` loop is used to compare the `key` with the elements to the left of it (`arr[j]`) and shift elements greater than the `key` to the right until a smaller element is found or the beginning of the array is reached. Within the `while` loop, the element at index `j` is moved one position to the right (`arr[j + 1] = arr[j]`) to make space for the `key` to be inserted at its correct position. Once the correct position for the `key` is found, it is placed in its sorted position in the array (`arr[j + 1] = key`). The `main` function serves as the entry point of the program. In the `main` function, the user is prompted to enter the number of elements they want to sort. The user inputs the value for `size`, indicating the number of elements to be sorted. The program dynamically allocates memory for an integer array named `arr` of size `size` using the `new` keyword. This ensures that the array can hold the specified number of elements.The program then prompts the user to enter the individual elements of the array. The user enters the elements one by one, which are stored in the dynamically allocated `arr` array. The program prints the unsorted array elements using a `for` loop and `cout`.The `insertionSort` function is called, passing the array `arr` and its size `size` as arguments, to sort the elements in ascending order. The program then prints the sorted array elements using another `for` loop and `cout`.In summary, the code demonstrates a simple implementation of the Insertion Sort algorithm, allowing the user to input any number of elements to be sorted. It showcases how the algorithm efficiently arranges the elements in ascending order by iteratively inserting elements into their correct positions.