

```
#include <iostream>

void countingSort(int arr[], int size) {
    int maxVal = arr[0];
    for (int i = 1; i < size; i++) {
        if (arr[i] > maxVal) {
            maxVal = arr[i];
        }
    }

    // 创建计数数组
    int* count = new int[maxVal + 1]();

    // 统计元素出现次数
    for (int i = 0; i < size; i++) {
        count[arr[i]]++;
    }

    // 根据统计结果重构有序序列
    int k = 0;
    for (int i = 0; i <= maxVal; i++) {
        while (count[i] > 0) {
            arr[k] = i;
            count[i]--;
            k++;
        }
    }

    // 释放计数数组的内存
    delete[] count;
}
```

这是计数排序函数，它接收一个整数数组和数组的大小作为参数。计数排序是一种非比较排序算法，它通过统计每个元素出现的次数，然后根据统计结果重构有序序列。首先，遍历数组找到最大值，确定计数数组的大小。然后，创建一个计数数组，并将每个元素出现的次数统计到计数数组中。接下来，通过遍历计数数组，将元素按照统计结果的顺序重构到原数组中，达到排序的目的。最后，释放计数数组的内存。

```
int main() {
    int arr[] = {4, 2, 2, 8, 3, 3, 1};
    int size = sizeof(arr) / sizeof(arr[0]);

    countingSort(arr, size);

    std::cout << "排序结果: ";
    for (int i = 0; i < size; i++) {
        std::cout << arr[i] << " ";
    }
    std::cout << std::endl;
}
```

```
    return 0;  
}
```

这是主函数，用于测试计数排序算法。它创建一个包含待排序元素的整数数组，并计算数组的大小。然后，它调用countingSort函数对数组进行排序。最后，它输出排序结果。

希望这些注释和解释能帮助您理解计数排序算法的思路和实现。