

```
#include <iostream>
#include <vector>
#include <algorithm>

void bucketSort(float arr[], int size) {
    // 创建桶
    std::vector<float> buckets[size];

    // 将元素放入对应的桶中
    for (int i = 0; i < size; i++) {
        int bucketIndex = size * arr[i];
        buckets[bucketIndex].push_back(arr[i]);
    }

    // 对每个桶内的元素进行排序
    for (int i = 0; i < size; i++) {
        std::sort(buckets[i].begin(), buckets[i].end());
    }

    // 合并桶中的元素得到有序序列
    int index = 0;
    for (int i = 0; i < size; i++) {
        for (float num : buckets[i]) {
            arr[index] = num;
            index++;
        }
    }
}
```

这是桶排序函数，它接收一个浮点数数组和数组的大小作为参数。桶排序通过将元素划分到对应的桶中，并对每个桶中的元素进行排序，然后将桶中的元素按照顺序合并得到有序序列。首先，创建一个存储桶的数组，桶的数量与数组大小相同。然后，根据元素的值计算桶的索引，将元素放入对应的桶中。接下来，对每个桶中的元素进行排序，这里使用了std::sort函数进行排序。最后，通过合并桶中的元素，按照顺序将元素放回原数组中，得到有序的数组。

```
int main() {
    float arr[] = {0.42, 0.32, 0.64, 0.12, 0.22, 0.11, 0.90};
    int size = sizeof(arr) / sizeof(arr[0]);

    bucketSort(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，用于测试桶排序算法。它创建一个包含待排序元素的浮点数数组，并计算数组的大小。然后，它调用bucketSort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桶排序算法的思路和实现。