

```
#include <iostream>

void selectionSort(int arr[], int size) {
    for (int i = 0; i < size - 1; i++) {
        int minIndex = i;

        // 在未排序部分中找到最小元素的索引
        for (int j = i + 1; j < size; j++) {
            if (arr[j] < arr[minIndex]) {
                minIndex = j;
            }
        }

        // 将最小元素交换到已排序部分的末尾
        int temp = arr[i];
        arr[i] = arr[minIndex];
        arr[minIndex] = temp;
    }
}
```

这是选择排序函数，它接收一个数组和数组的大小作为参数。选择排序通过不断选择未排序部分的最小元素，并将其交换到已排序部分的末尾来实现排序。外部循环控制已排序部分的末尾位置，从数组的第一个元素开始到倒数第二个元素。内部循环在未排序部分中查找最小元素的索引，并将最小元素的索引保存在minIndex变量中。然后，将最小元素与已排序部分的末尾元素交换位置。通过多次的选择和交换操作，将最小的元素逐步放置在已排序部分的末尾。

```
int main() {
    int arr[] = {64, 34, 25, 12, 22, 11, 90};
    int size = sizeof(arr) / sizeof(arr[0]);

    selectionSort(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，用于测试选择排序算法。它创建一个包含待排序元素的数组，并计算数组的大小。然后，它调用selectionSort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选择排序算法的思路和实现。