# 计数排序的代码实现

## 1. 实战版本（更通用，以下代码可直接运行）

```c
#include <stdio.h>
#include <stdlib.h>

void counting_sort(int A[], int n) {
    // 1. 找到数组 A 中的最大值和最小值
    int max = A[0];
    int min = A[0];
    for (int i = 1; i < n; i++) {
        if (A[i] > max) {
            max = A[i];
        }
        if (A[i] < min) {
            min = A[i];
        }
    }

    // 2. 创建计数数组
    int k = max - min + 1;
    int* C = (int*)calloc(k, sizeof(int));  // 动态分配内存

    // 3. 统计每个元素出现的次数
    for (int i = 0; i < n; i++) {
        C[A[i] - min]++;
    }

    // 4. 将计数数组转化为累加数组
    for (int i = 1; i < k; i++) {
        C[i] += C[i - 1];
    }

    // 5. 构建输出数组
    int* B = (int*)malloc(n * sizeof(int));  // 动态分配输出数组
    for (int i = n - 1; i >= 0; i--) {  //从后往前处理，确保稳定性
        B[C[A[i] - min] - 1] = A[i];
        C[A[i] - min]--;
    }

    // 6. 将输出数组B拷贝回原数组A
    for (int i = 0; i < n; i++) {
        A[i] = B[i];
    }

    // 7. 释放内存
    free(C);
    free(B);
}
```

```
int main() {
    int A[] = {4, 2, 2, -3, 5, 0, 3, 3, 1};
    int n = sizeof(A) / sizeof(A[0]);

    counting_sort(A, n);

    printf("排序后的数组: ");
    for (int i = 0; i < n; i++) {
        printf("%d ", A[i]);
    }
    printf("\n");

    return 0;
}
```

## 2. 教学用代码（更简洁易懂，改写自《算法导论》）

```
// 计数排序
void CountSort(int A[], int B[], int n, int k){
  int i,C[k];              //辅助数组C的长度取决于待排序元素取值范围[0, k)
  for(i=0;i<k;i++)         //初始化计数数组
    C[i]=0;
  for(i=0;i<n;i++)         //Step1：遍历待排序数组，统计每个关键字的出现次数
    C[A[i]]++;
  for(i=1;i<k;i++)         //Step2：再次处理辅助数组，统计不大于 i 的元素个数
    C[i]=C[i]+C[i-1];      //C[i]保存的是小于或等于 i 的元素个数
  for(i=n-1;i>=0;i--){     //Step3：利用辅助数组C实现计数排序（从后往前处理）
    C[A[i]]=C[A[i]]-1;
    B[C[A[i]]]=A[i];  //将元素 A[i]放在输出数组 B[]的正确位置上
  }
}
```