

测试思路

我采用以下测试策略：

1. **测试用例 1**：常规输入，包含重复数字
 - 检查排序功能是否正常工作
 - 检查去重功能是否正常工作
2. **测试用例 2**：空输入
 - 确保程序能够妥善处理没有数据的情况
3. **测试用例 3**：只有一个数字
 - 验证边界情况处理

编译与运行

.\文件夹下在控制台中输入如下指令：

```
1. cd build
2. cmake -G "MinGW Makefiles" ..
3. cmake --build .
4. .\HW9.exe
```

测试结果示例

程序运行后预期输出：

```
5. === 运行测试用例 ===
6. 测试用例 1 结果：
7. 使用 sort 函数排序结果(不去重)：
8. 1 3 3 5 5 8 9
9. 使用 set 排序结果(去重)：
10. 1 3 5 8 9
11.
12. 测试用例 2 结果（空输入）：
13. 没有数据可排序！
14. 没有数据可排序！
15.
16. 测试用例 3 结果（单个数字）：
17. 使用 sort 函数排序结果(不去重)：
18. 42
```

```
19. 使用 set 排序结果(去重):
20.42
21.
22.=== 开始实际程序 ===
23.请输入要处理的整数个数: 5
24.请依次输入 5 个整数:
25.8 2 5 2 9
26.使用 sort 函数排序结果(不去重):
27.2 2 5 8 9
28.使用 set 排序结果(去重):
29.2 5 8 9
```

程序运行实际结果

```
PS C:\Files\OOP\HW9\build> .\HW9.exe
=== 运行测试用例 ===
测试用例1结果:
使用sort函数排序结果(不去重):
1 3 3 5 5 8 9
使用set排序结果(去重):
1 3 5 8 9

测试用例2结果 (空输入) :
没有数据可排序!
没有数据可排序!

测试用例3结果 (单个数字) :
使用sort函数排序结果(不去重):
42
使用set排序结果(去重):
42

=== 开始实际程序 ===
请输入要处理的整数个数: 5
请依次输入5个整数:
8 2 5 2 9
使用sort函数排序结果(不去重):
2 2 5 8 9
使用set排序结果(去重):
2 5 8 9
```

与预期相符。