



排序库函数

在 C++ 语言里，已经提供了排序效率很高的内置排序库函数 `sort()`。

使用这个函数，需要

```
1. #include<algorithm>
```

这个函数的第一种调用方式，包含两个参数：

```
1. //将数组元素 a[m]..a[n-1]从小到大排序
2. sort(a+m, a+n);
```

这个函数的第二种调用方式，包含三个参数：

```
1. //将数组元素 a[m]..a[n-1]按照 cmp 函数定义的比较规则，进行排序
2. //cmp 是对两个元素进行比较，返回值为 bool 类型的函数。返回 true 表示第一个元素在第二
   个元素之前，否则返回 false
3. sort(a+m, a+n, cmp);
```



使用库函数排序

【问题描述】

从键盘读入 10 个数，然后从小到大输出。

【分析】

使用 `sort(a+n, a+m)` 进行排序。

【参考程序】

```
1. #include<iostream>
2. #include<algorithm>
3. using namespace std;
4. int main()
5. {
6.     int a[10];
7.     for (int i=0; i<10; i++)
8.     {
9.         cin >> a[i];
10.    }
11.    sort(a, a+10);
12.    for (int i=0; i<10; i++)
13.    {
14.        cout << a[i] << ' ';
15.    }
16.    return 0;
17. }
```



使用库函数自定义排序

【问题描述】

从键盘读入 10 个数，然后从大到小输出。

【分析】

使用 `sort(a+n, a+m, cmp)` 进行排序。

【参考程序】

```
1. #include<iostream>
2. #include<algorithm>
3. using namespace std;
4. bool cmp(int a, int b)
5. {
6.     return a>b;
7. }
8. int main()
9. {
10.     int a[10];
11.     for (int i=0; i<10; i++)
12.     {
13.         cin >> a[i];
14.     }
15.     sort(a, a+10, cmp);
16.     for (int i=0; i<10; i++)
17.     {
18.         cout << a[i] << ' ';
19.     }
20.     return 0;
21. }
```