

简答题

- (1) C++ 中为什么要使用模板?
- (2) C++ 中模板有哪两种形式, 有什么区别?

编程题

(1) 用函数模板, 实现一个 `sort(array, length)` 函数, 该函数可以对长度为 `length` 的数组 `array` 进行从小到大的排序。要求能够对所有基本类型适用。

调用示例:

```
int int_array = [4, 2, 1, 3];
sort(int_array, 4);
for(int i = 0; i < 4; i++)
    std::cout << int_array[i] << endl;
```

```
char char_array = ['c', 'd', 'a'];
sort(char_array, 3);
for(int i = 0; i < 3; i++)
    std::cout << char_array[i] << endl;
```

```
double double_array = [1.2, 0.3, 5.8, 1.9, 2.0];
sort(double_array, 5);
for(int i = 0; i < 5; i++)
    std::cout << double_array[i] << endl;
```

结果:

```
1
2
3
4
a
c
d
0.3
1.2
1.9
2.0
5.8
```

提交要求:

- 1 提交两个源码文件, `sort.h` 和 `sort.cpp` 直接打包成 `zip` 压缩格式的压缩包。不要添加任何其他目录。
- 2 实现代码必须写在 `cpp` 文件中, 声明代码必须写在 `h` 头文件中, 文件编码格式为 `utf-8`。

- 3 请严格按照给定的接口进行编码，否则无法调用测试用例。
- 4 提交的源码文件中，不能包含 main 函数，否则无法编译通过。

(2) 队列 Queue 是一种先进先出的数据结构，用类模板实现一个队列类，假设元素类型为 T, Queue 类的接口要求如下：

```
Queue(); //无参构造函数，初始化队列为空，size为零
~Queue(); //析构函数，删除所有元素，回收空间
void push(T e); //元素e入队操作，在队尾插入
T pop(); //出队操作，在队头删除，返回出队的元素
T back(); //返回队尾元素
T front(); //返回队头元素
bool isEmpty(); //队列为空返回true，否则返回false
size_t size(); //返回队列长度
void show(); //显示队列元素，以空格分隔元素，在最后添加换行符。
```

调用示例：

```
Queue<int> q1;
q1.push(1);
q1.push(2);
q1.push(3);
q1.push(4);
q1.pop();
q1.show();
Queue<string> q2;
q2.push("test");
q2.push("queue");
q2.push("template");
cout << q2.front() << endl;
q2.pop();
q2.show();
```

结果：

```
2 3 4
test
queue template
```

注意事项：

要求使用类模板，支持所有的基本的数据类型，请注意 Queue 类的声明，不要使用 stl。

提交要求：

- 1 提交两个源码文件，Queue.h 和 Queue.cpp 直接打包成 zip 压缩格式的压缩包。不要添加任何其他目录。

- 2 实现代码必须写在 `cpp` 文件中，声明代码必须写在 `h` 头文件中，文件编码格式为 `utf-8`。
- 3 请严格按照给定的接口进行编码，否则无法调用测试用例。
- 4 提交的源码文件中，不能包含 `main` 函数，否则无法编译通过。