

## 编程题

(1) 实现一个 merge 函数，该函数可以将两个有序数组合并成一个新的有序数组，**要求能够对所有基本类型适用**。(40 分)

接口要求：

//返回 array\_1 和 array\_2 的合并结果，需要在函数内用 new 申请内存空间。

```
T* merge(T* array_1, size_t size_1, T* array_2, size_t size_2);
```

调用示例：

```
int int_array_1[3] = {1, 2, 4};
int int_array_2[1] = {3};
int *int_res = NULL;
int_res = merge(int_array_1, 3, int_array_2, 1);
for(int i = 0; i < 4; i++)
    std::cout << int_res[i] << endl;

char char_array_1[2] = {'c', 'd'};
char char_array_2[2] = {'a', 'z'};
char *char_res = NULL;
char_res = merge(char_array_1, 2, char_array_2, 2);
for(int i = 0; i < 4; i++)
    std::cout << char_res[i] << endl;

double double_array_1[5] = {0.3, 1.2, 1.9, 2.0, 5.8};
double double_array_2[2] = {0.1, 2.8};
double *double_res = NULL;
double_res = merge(double_array_1, 5, double_array_2, 2);
for(int i = 0; i < 7; i++)
    std::cout << double_res[i] << endl;
```

结果：

```
1
2
3
4
a
c
d
z
0.1
0.3
```

1.2

1.9

2.0

2.8

5.8

提交要求：

- 1 提交一个源码文件，merge.h 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 实现和声明代码都写在 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) 实现复数的四则运算。要求重载  $= + - * /$  6 个操作符。(40 分)

接口要求：

```
class Complex {
private:
    double re; //实部
    double im; //虚部

public:
    //TODO: 构造函数，参数形式可以是：1) 不带参数，即 re = 0, im
    = 0; 2) 一个参数，即 re = 参数, im = 0; 3) 两个参数，即
    re = 参数1, im = 参数2
    //TODO: 析构函数

    //TODO: 重载运算符 =
    //TODO: 重载运算符 +
    //TODO: 重载运算符 -
    //TODO: 重载运算符 *
    //TODO: 重载运算符 /
    //TODO: 重载 << , 输出 re 和 im, 以空格分隔 (保留2位小数)。
}
```

调用示例：

```
Complex x(1);
Complex y(3, 4);
Complex z;
z = x + y;
cout << z;
cout << x - y;
```

```
cout << x * y;  
cout << x / y;
```

结果:

```
4.00 4.00  
-2.00 -4.00  
3.00 4.00  
0.12 -0.16
```

提交要求:

- 1 提交两个源码文件, Complex.h 和 Complex.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 实现写在 cpp 文件中, 声明写在 h 文件中, 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。

(3) 在 STL 中, 迭代器 (iterator, 也称作智能指针) 是一个抽象的设计理念, 通过迭代器可以在不了解容器内部原理的情况下遍历容器。除此之外, stl 中迭代器一个重要作用就是作为容器 (vector, list, stack 等) 和算法 (sort 等) 的粘结剂, 只要容器提供迭代器接口, 同一套算法代码可以利用在完全不同的容器中, 这是抽象思想的典型应用。实现一个简单的数组迭代器。(20 分)

假设数组类型为 T, Iterator 的接口要求如下:

```
T *ptr; //成员变量  
Iterator(T *p = 0) : ptr(p) {}; //构造函数。  
~Iterator();  
//TODO: 重载 + 操作, 与普通 size_t 的加法  
//TODO: 重载 - 操作, 与普通 size_t 的减法  
//TODO: 重载 ++ 操作, 包括 i++ 和 ++i  
//TODO: 重载 -- 操作, 包括 i-- 和 --i  
//TODO: 重载 = 操作  
//TODO: 重载 == 操作  
//TODO: 重载 != 操作  
//TODO: 重载 *, 返回当前迭代器指向元素的引用。
```

调用示例:

```
int int_array[4] = {1,2,3,4};  
Iterator<int> i_begin(int_array);  
Iterator<int> i_end(int_array + 4);  
for(Iterator<int> i = i_begin; i != i_end; i++) {  
    cout << *i << endl;  
}
```

```
string string_array[2] = { 'world', 'hello' };
Iterator<string> s_begin(string_array);
Iterator<string> s_end(string_array + 2);
for(Iterator<string> i = s_end - 1; i != s_begin; --i){
    cout << *i << endl;
}
```

结果：

```
1
2
3
4
hello
```

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 Iterator 类的声明，不要使用 stl。

提交要求：

- 1 提交一个源码文件，Iterator.h 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 实现和声明都写在 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。