

# 直线相交

## 直线相交

[刷新 ↻](#)

### 题目说明

在二维坐标系中，每一个点可以用  $(x, y)$  表示，其中  $x$  表示横坐标， $y$  表示纵坐标。

每一条直线可以使用  $y = kx + b$  的形式表示，其中  $k$  为斜率， $b$  为截距。

请设计一个点类 `Point` 和一个直线类 `Line`，要求：

需要使用类模板的方式。使得既能够创建  $x, y$  为整数的 `Point` 对象， $k, b$  为整数的 `Line` 对象；也能创建  $x, y$  为实数的 `Point` 对象， $k, b$  为实数的 `Line` 对象。

`Point` 通过传入  $x, y$  进行初始化；`Line` 可以通过同时传入  $k, b$  进行初始化，也可以仅传入  $k$  进行初始化（此时  $b$  默认为 0）。

两个 `Line` 对象间应该能够比较大小。我们定义：两条直线的大小完全取决于二者的斜率大小。

`Point` 具有如下三个接口：

`void move(T dx, T dy)`：让该点的横坐标移动  $dx$  距离，纵坐标移动  $dy$  距离。

`void show()`：输出点的横坐标  $x$  和纵坐标  $y$ ，格式为  $(x,y)$ ，且最后添加一个换行符。

`bool check(Line<T> const& l)`：传入一个 `Line` 对象，判断点是否在这条直线上。

`Line` 具有如下两个接口：

`bool intersect(const Line&)`：传入另一个 `Line` 对象，判断两条直线是否存在交点。如果两条直线重合，也认为存在交点。

`void show()`：输出直线的  $k$  和  $b$ 。其中  $k$  和  $b$  之间使用逗号隔开，且最后添加一个换行符。

此外，两个 `Line` 对象间应该能够比较大小，**这意味着你需要在 `Line` 类中重载 `<` 运算符**

在本题中，我们假设  $x, y, k, b, dx, dy$  的数据类型相同。对于实数类型的数据，当两个数差的绝对值小于  $1e-6$  时，我们认为两个数相等。我们对输出格式的有效数字位数不做特殊要求。

一种使用 `Line` 类的示例为：

```
#include<iostream>
#include"Line.cpp"
#include"Point.cpp"
using namespace std;

int main() {

    Point<int> p1(2,4);
    p1.move(3,2);
    p1.show();
    // 此处应该输出(5,6)

    Line<int> line1(2);
    Line<int> line2(2, 4);
    cout << line1.intersect(line2) << endl;
    // 此处应输出 0

    Line<int> line3(3);
    Line<int> line4(2, 6);
    if (line3 < line4) line3.show();
    else line4.show();
    // 此处应输出 2,6

    Point<int> p2(4,12);
    cout<<p2.check(line2)<<endl;
    //此处应该输出1
    return 0;
}
```

### 提交格式

- 你需要提交多个文件，包含 `Line.h`, `Line.cpp`, `Point.h` 以及 `Point.cpp` 文件；不需要上传 `main.cpp` 和 `Makefile` 文件。

- 你应该将你的文件打包成一个zip压缩包并上传。**注意：你的文件应该在压缩包的根目录下，而不是压缩包的一个子文件夹下。**评测时，OJ会将提供的main.cpp贴入你的目录下进行编译并执行。

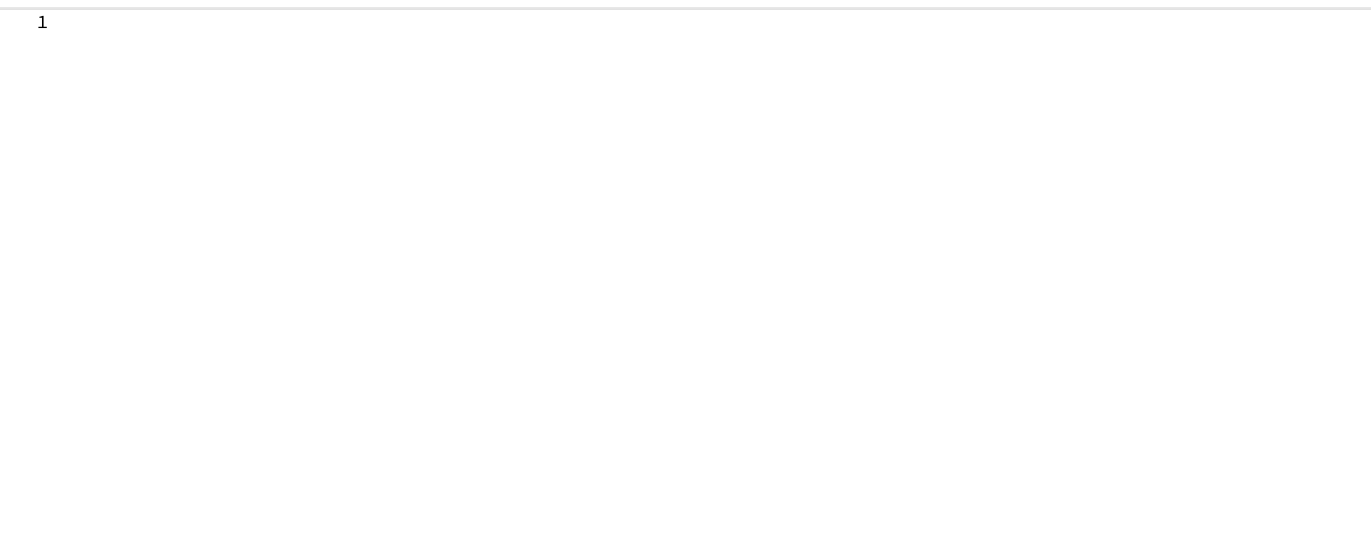
## 语言和编译选项

#	名称	编译器	额外参数	代码长度限制
0	oop_custom	make		1048576 B

递交历史

递交答案

语言和编译选项 `oop_custom`



提交

文件请拖入编辑器中，或 

上传文件

提交

文件请拖入编辑器中，或 

上传文件