

# 暑假实习那段代码

害怕自己哪天突然因为泄露大比例尺地形图数据被抓了，示例数据里面我全部都加了几千米的偏移，但是相对位置没有发生变化。

## 操作步骤

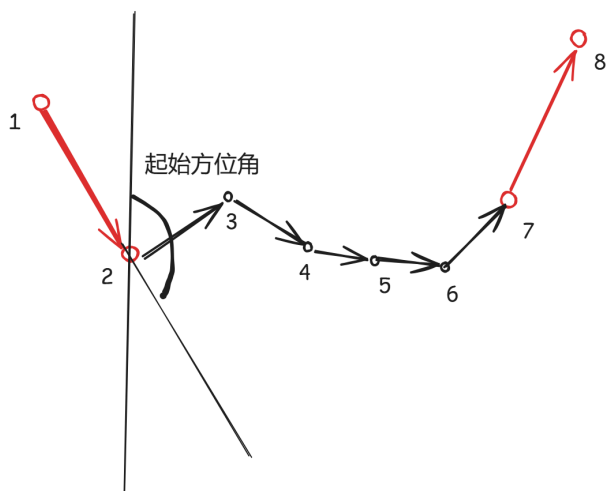
输入1、2点的坐标，确定下起始方位角。终边一样。

然后是依次输入“2”站“3”站……“7”站的左角。也就是总共(n-2)站，总共只测这些站。  
(数据记录表应该点号一栏是1 2 3 4 5 6 7 8，不要因为1和8没设站就不填)

左角输入完毕后，输入 **2227** 这个数字，是结束标志（也可以改成你自己的学号后四位，或者你喜欢的数字）

下面输入距离，只需要(转角个数-1)个，已知边的边长其实是不用测的。也就比如第一张图，1 2和7 8之间的距离是不用输入的。

## 注意事项



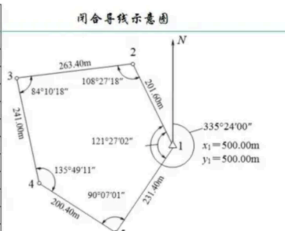
如图，注意不要把起始方位角当成了大于180度的那个。

你要是先输入2，然后1，得到的方位角是接近360度的那个大角，是现在这个起始方位角+180°。

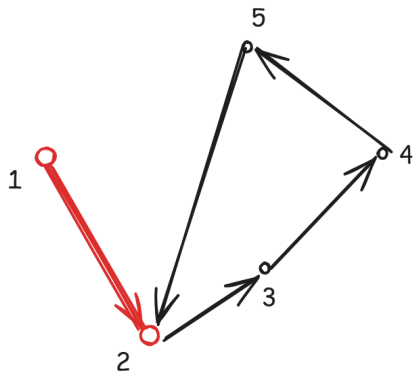
计算方位角我用的是N E分别末-初，然后用 `atan2` 函数得出象限角，因为 `atan` 不会区分象限。

先输入1的坐标，然后是2的坐标。同理，先输入7的坐标，然后是8的坐标。

就是输入顺序与行进方向统一

[illegible]

这里的从1 2开始，最后到1 2结束，就是先输入1的坐标，然后是2的坐标。同理，先输入1的坐标，然后是2的坐标。



这样的闭合导线和上面的不一样，从1 2开始，但结束是2 1，注意行进方向。

里面写的很混乱，但是我实在懒得改了。

## 一些很混乱的点（没改）

- ☐ processing.hpp，这个是主要的头文件。里面好几个函数是重复的，比如 `DMS(double radians);` // 将弧度转化为DMS 这个函数根本没用到。
- ☒ 在强调了按照行进方向输入坐标之后 `cout << "输入坐标起算点(第一站)的N E坐标" << endl;` 和 `cout << "输入最后一站的前视点的N E坐标" << endl;` 其实已经没用了，这是当时我没意识到用 `atan` 而不用 `atan2` 可能会有危险的情况，因为前者不会判断象限。
- ☐ 变量名非常混乱，比如v1 v2 v3，现在这些我自己都看不懂是什么含义，能运行也真是个奇迹。
- ☒ 一些函数名起的很不合理，不够直观。比如 `rechange`，我当时想的是弧度往回转化为DMS，所以用了 `rechange` 这个单词，体现了来源，但是没体现目的。
- ☐ 找出v2个最大的数的下标，并存在一个数组 这段代码应该用函数更有条理，直接写实在是太长了。
- ☒ 坐标增量的那个分配改正数算法可能不对，因为可能计算闭合差用的可能是最后一站的坐标，而不是最后一个前视点的坐标。而这个算法用的是后者，好奇怪呀，我也不确定到底是哪个。大概是不需要最后一个前视点的坐标，改这个比较麻烦，但已经改过来了。需要把有 `length` 的for循环，循环次数都改成 `count-1` 反号分配闭合差这个算法大概在 `导线测量.cpp` 的222行附近，需要的话，自己把数减1。
- ☒ 215行和245行附近我留了个 `TODO`，可能计算方法还有错误。
- ☐ 分配改正数那个算法简直一坨
- ☐ 分配改正数那个算法可能有错， `-(length[i] / lengthsum) * esum` 可能不是根据长度直接加权分配，而是根据增量比例分配
- ☐ 需要更有条理，比如说一些该用函数的地方给独立出来，要不然太乱了。

