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

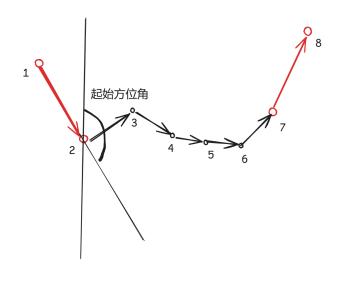
输入1、2点的坐标,确定下起始方位角。终边一样。

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12345678,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,12和78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°。

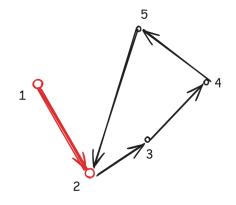
计算方位角我用的是N E分别末-初,然后用 atan2 函数得出象限角,因为 atan 不会区分象 限。

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

序号	点号或 点名	观測左角值 (d.ms)	改正后左角值 (***)	坐标方位角 (***)	边长(m)	计算坐标增量(m)		改正数(m)		改正后坐标增量(m)		平差后坐标(m)		备注	a k z w z k a	
						÷χ	-y	X	Y	÷χ	-y	X	Y	會注	闭合导线示意图	
1	1										*****	1500000.000	2000000.000		, N	
2	2	108.2718	108*27'16"	335.2400	201.600	183.302	-83.922	0.001	-0.003	183.303	-83.925	1500183.303	1999916.075		2	
3	3	84.102	84*10*18*	263*51*16*	263.410	-28.200	-261.896	0.001	-0.004	-28.199	-261.900	1500155.104	1999654.175		3 263.40m	
4	4	135,4911	135*49'09"	168*01'33"	241.003	-235.759	50.001	0.001	-0.004	-235.758	49.997	1499919.346	1999704.172		04010/10"	
5	5	90.0621	90*06*19*	123*50'42"	200.360	-111.590	166.409	0.001	-0.003	-111.589	166.406	1499807.757	1999870.578		- 17 8+1018 /B	
6	1	121.2702	121*26'60"	33*57'00"	231.750	192.242	129.426	0.001	-0.003	192.243	129.422	1500000.000	2000000.000		335°24'00"	
7	2	121.2702	121 20 00	335*24'00"								1000000.000	200000.000		131022001	
-	2														$x_1 = 500.00$ m $y_1 = 500.00$ m	
8															135°49′11″	
9															40)	
10															300 90*07*01" 55	
11															300.400	
12																

这里的从12开始, 最后到12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12开始,但结束是21, 注意行进方向。

里面写的很混乱, 但是我实在懒得改了。

<u></u> j	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,这是当时我没用意识到用 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。
<u> </u>	一些函数名起的很不合理,不够直观。比如rechange,我当时想的是孤度往回转化为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行附近我留了两个 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。