

1. Dev不输出有可能是修改代码之后没有重新编译
2. 内部类和外部类没有继承或friend关系
3. 在对关联容器自定义比较规则的时候要在重载<之后写const, 因为set或map内部用了const
4. iterator末尾是or不是er
5. 0207
  - (1) 不用B::Fun就不会输出B::Fun了
  - (2) 多态除了使用基类指针, 也可以用基类引用触发
6. 0212
  - (1) 有const限定符的函数算重载
7. 0213
  - (1) Copy对应ostream\_iterator<int> itr(cout, "")
  - (2) 不能用char\*初始化char a[100], 要循环遍历
8. 0303
  - (1) friend istream也有两个参数, 且变量要写引用否则读不进来
  - (2) 初始化static也要写类型名: int
9. 0304
  - (1) 根本不用Print, 直接重载accumulate
10. 0305
  - (1) T生成数组同样不能用变量初始化数组大小
11. 0307
  - (1) 如何处理函数指针: template<class Tf> Tf sum(int a, int b, Tf (\*f)(T))
12. 0309
  - (1) return x % y == 0 ? y : gcd(y, x % y);
13. 0311
  - (1) Set和map的比较函数一定是const struct, 不能是函数
  - (2) 多组数据在N之外什么都不要写!
14. 0505
  - (1) Lambda嵌套, 且内层要写[=]
15. 0506
  - (1) 如果输出的答案无法直接操作, 可以取出它的指针
  - (2) 如果不知道总共几次, 可以每一次都算出结果。如算平均值, 不知道什么时候要除以n, 可以每次都算出来临时的平均数, 保存下来除以n之前的数下一次再用
16. 0507
  - (1) 不要用优先队列, 不能修改其中元素, 用map
  - (2) 用map时要注意是否有重复元素, 如果有就用multimap
17. 0508
  - (1) 重载加号实现A于A\*类型的加法
- 18.

