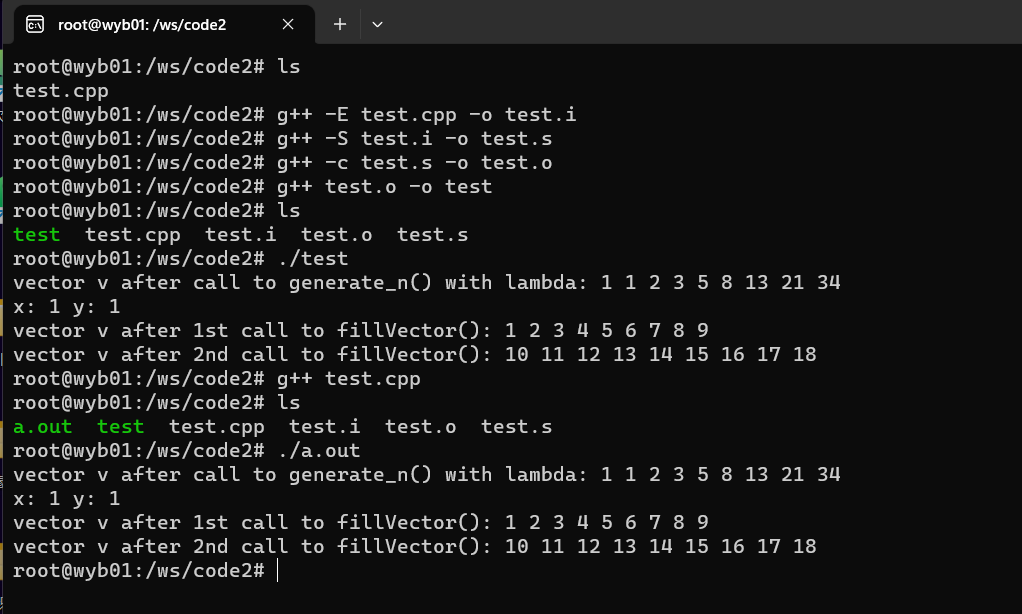
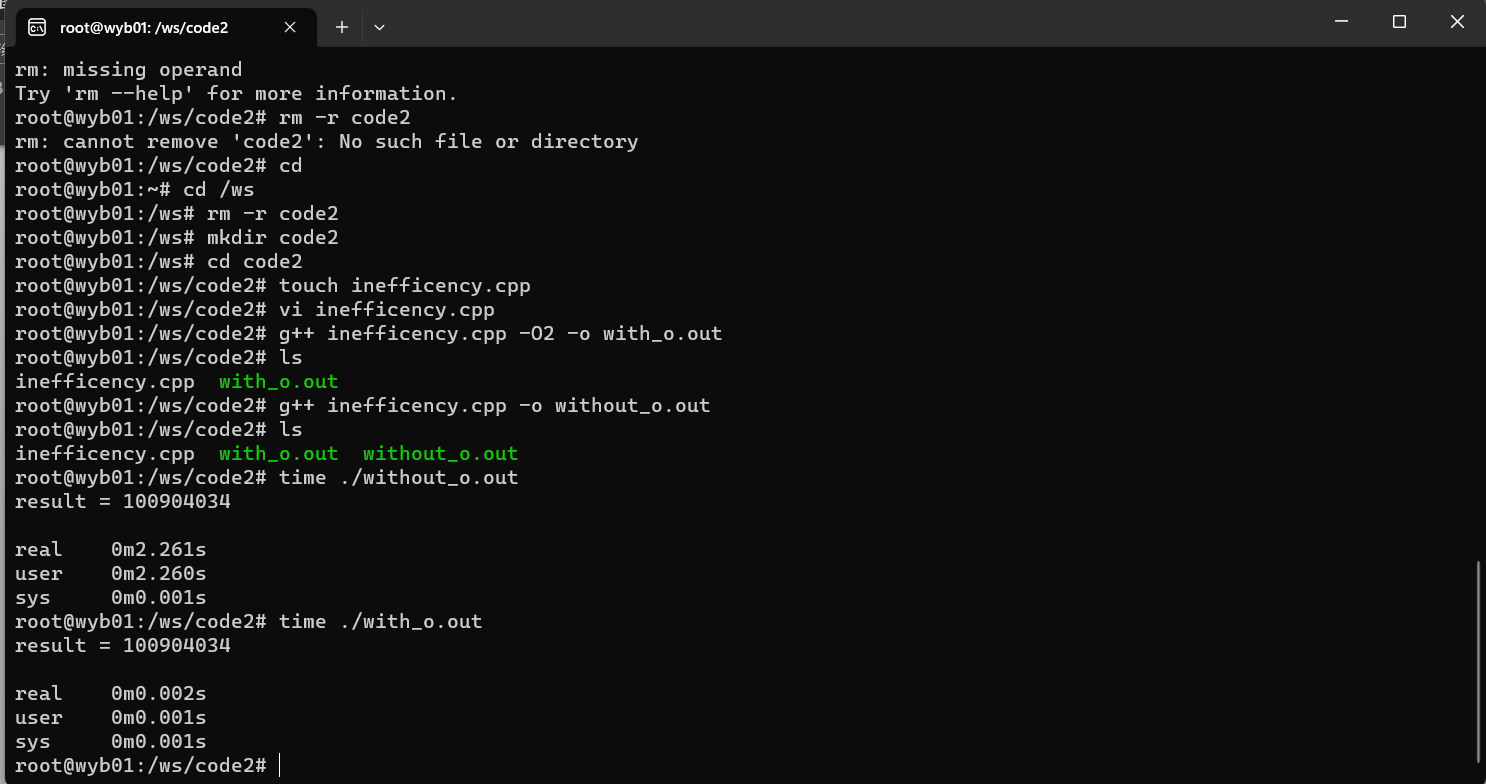
第一第二大题因docker故障采用远程ssh连接Ubuntu虚拟机

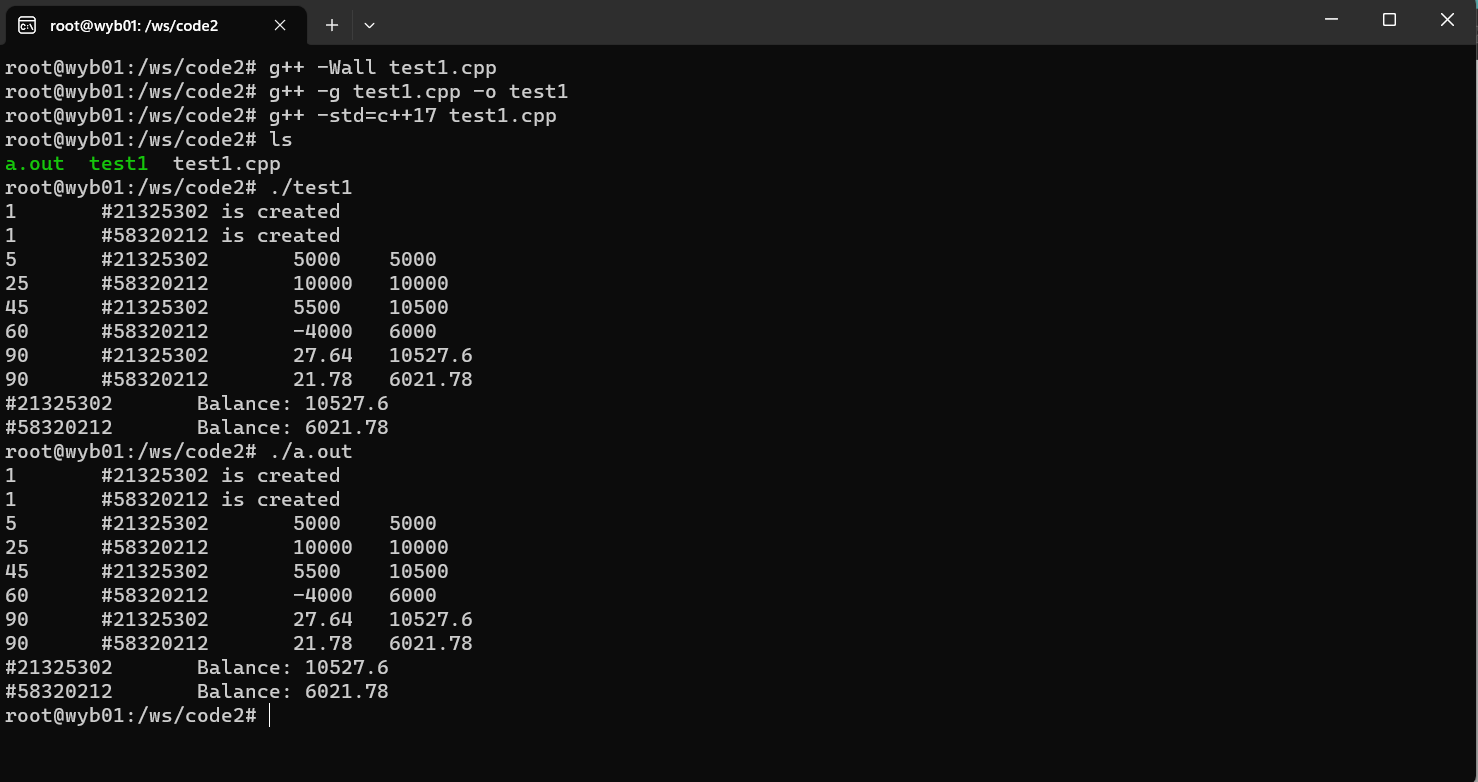
三．1.单步编译与分步编译

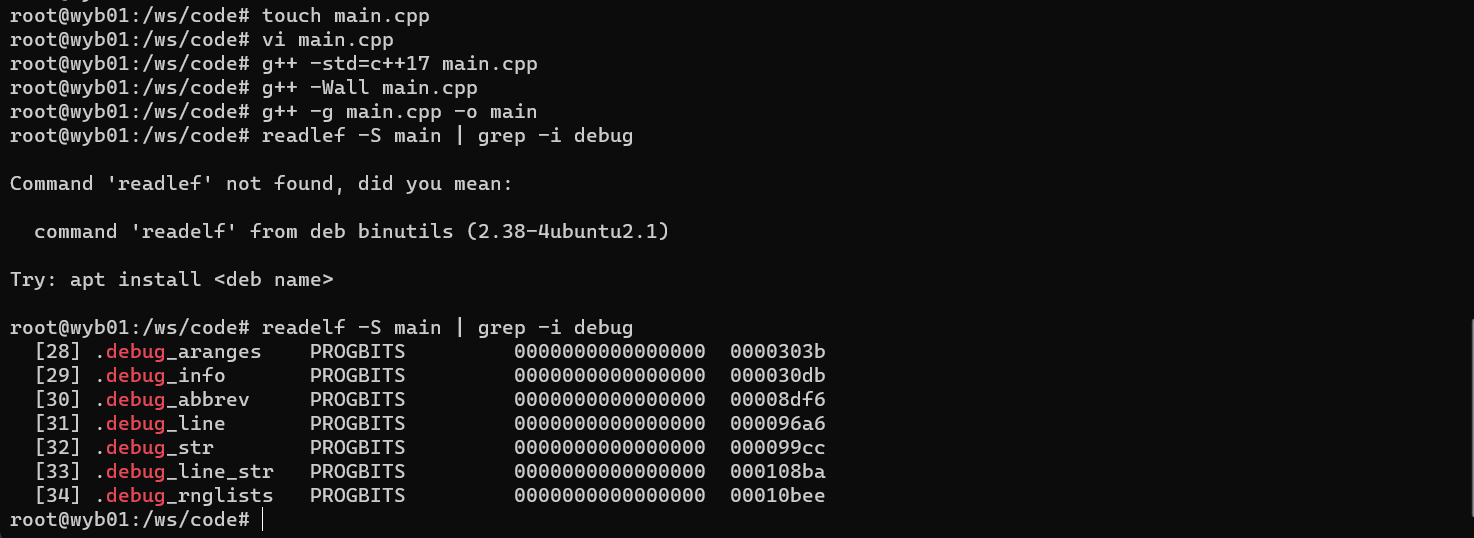


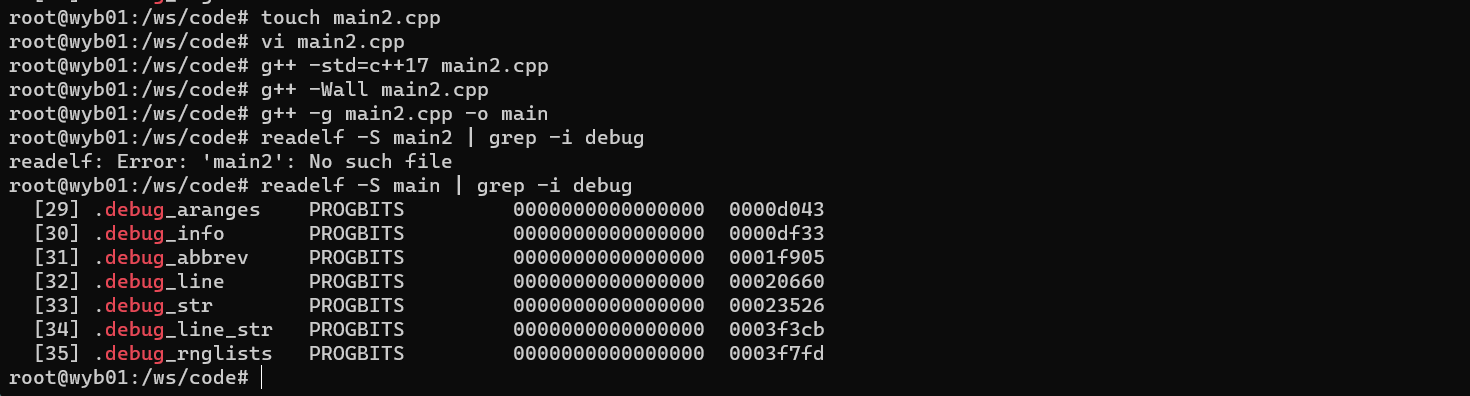
2. inefficency.cpp 优化编译与非优化编译的执行信息



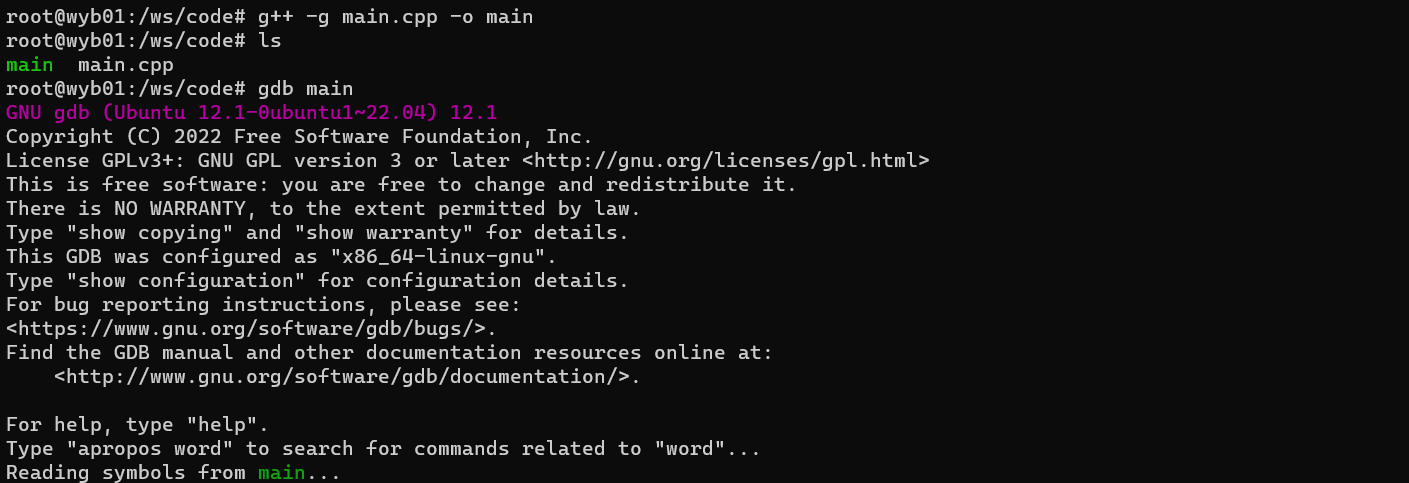
3.不同g++参数编译，并打印最终结果







1. 1.追踪前后两次调用函数 sumOfSquare() 时函数调用的栈帧与层级关系
   1. .编译+gdb调试

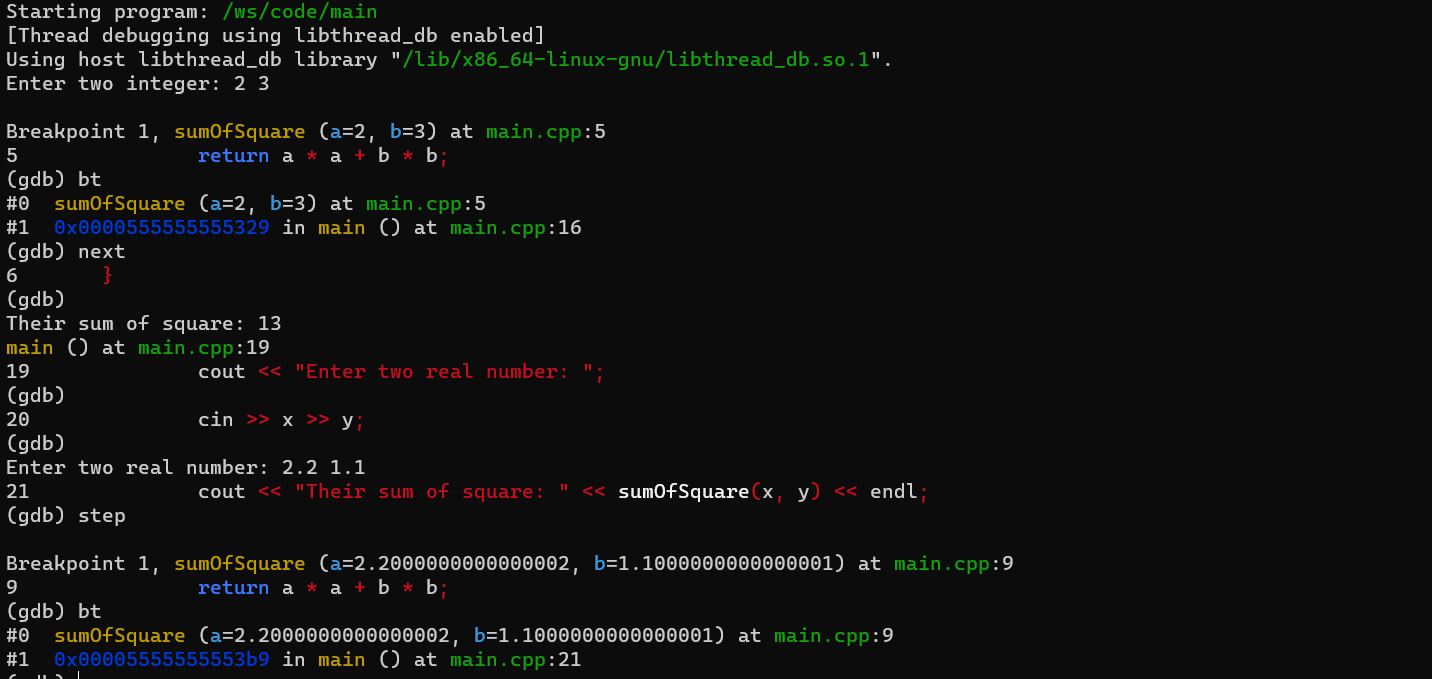


* 1. .设置断点，运行

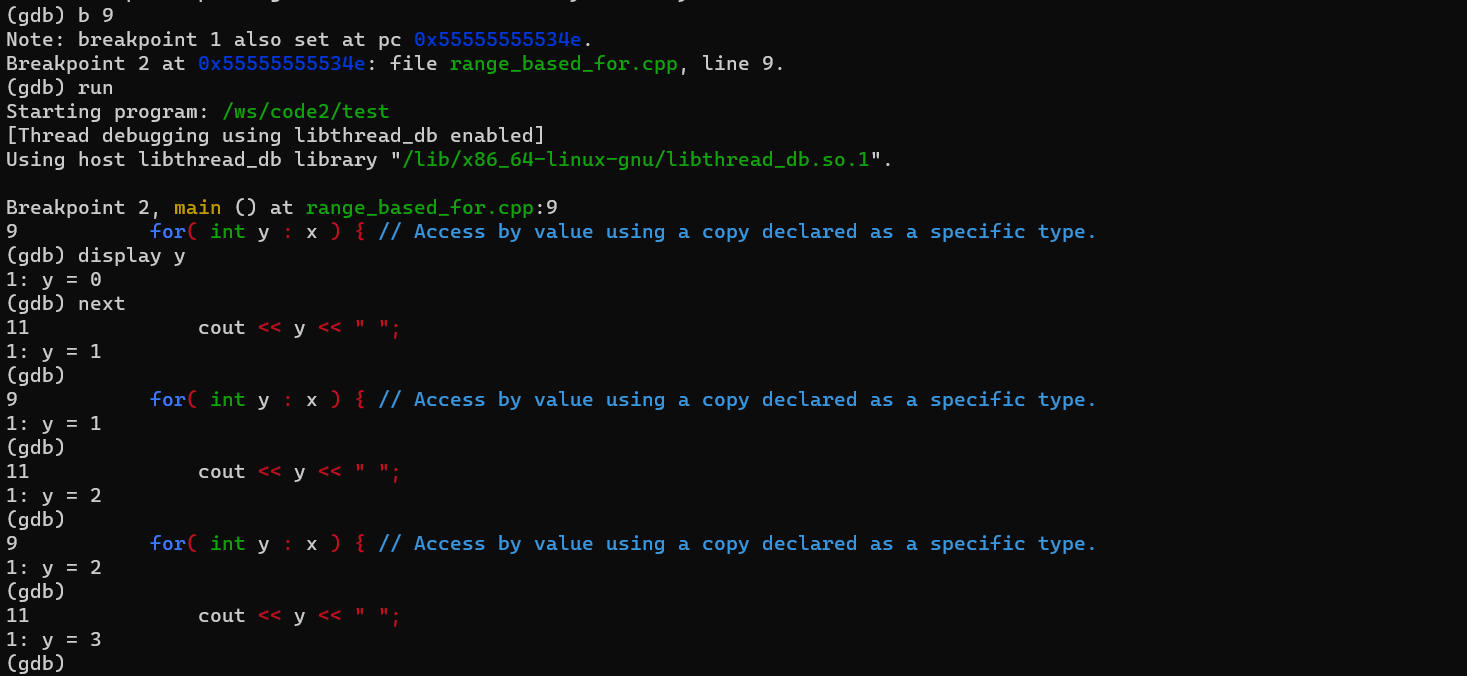
图形用户界面, 文本, 应用程序

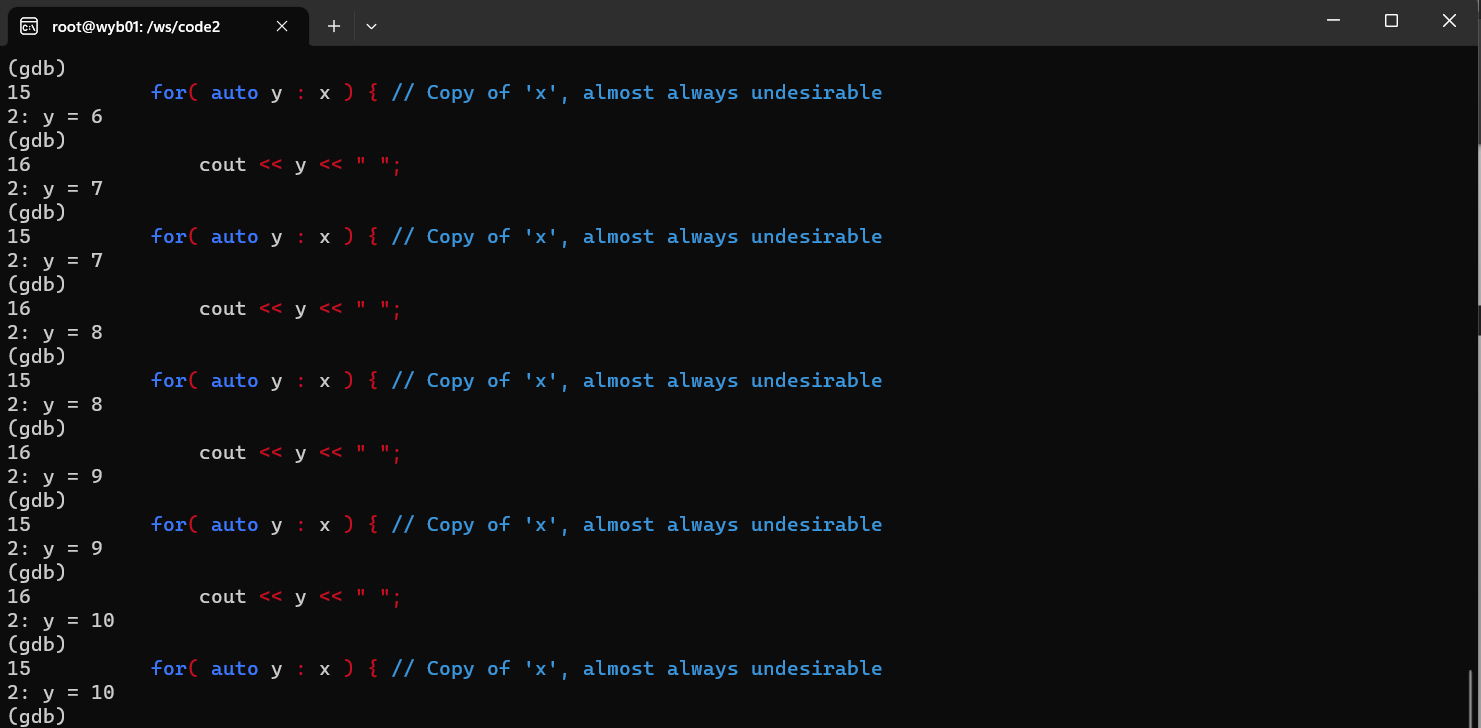
描述已自动生成

* 1. .分别在两个函数调用时查看栈帧与层级关系(backtrace 命令)

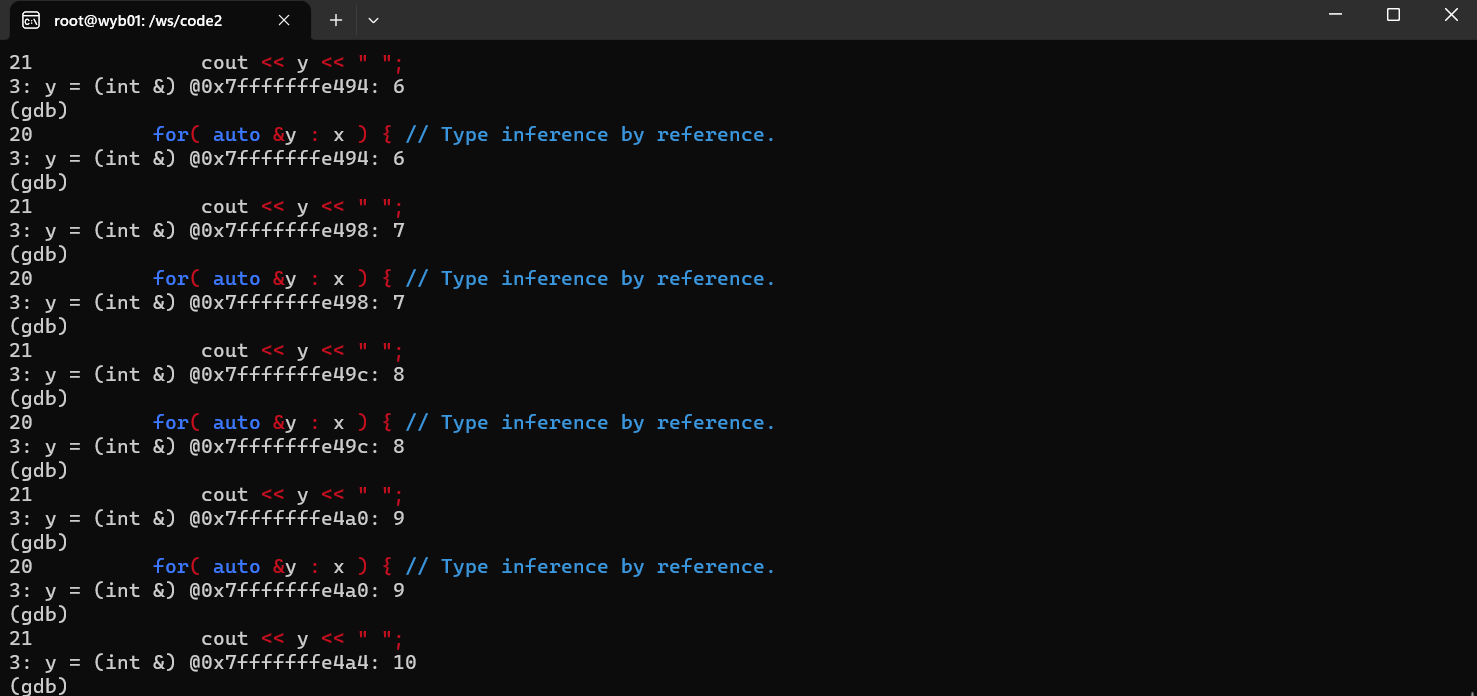


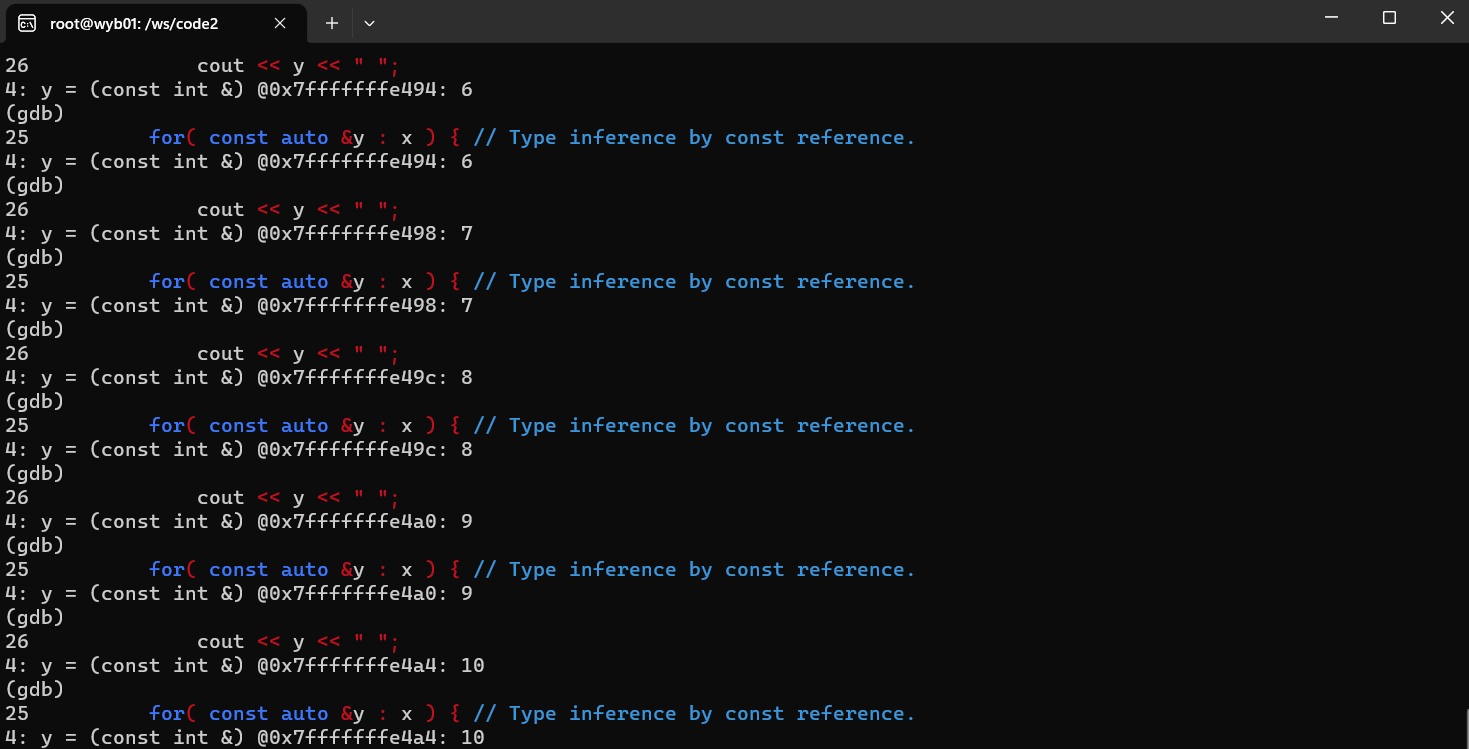
2.1)设置断点监视y的变化（四个循环）



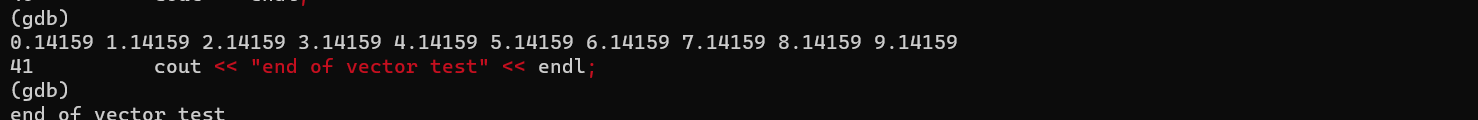


引用形式传递：





2）.打印出来j的值



Vector<double> v容器敬礼for循环在尾部进行push\_back数值是i+0.14159（i由0到9），

然后进行基于范围的for循环，传入每个元素的类型是const auto&类型的，所以每个元素的内容是只读的，最终打印出来j的值。

3.结论：在此问题中enum没有地址，const和define有地址



设置断点之后，可以发现p在Pre-processing中被替换成了“hello”，所以define p打印的值有地址。

c.NUM是static const int类型，可以发现打印出来了地址。

而C：：NUM1是enum类型，并没有打印出来地址Can't take address of "C::NUM1" which isn't an lvalue.

故得到上述结论