

济南历城二中信息学奥林匹克联赛校内模拟赛

(yLOI2020)

选手须知

(请选手务必仔细阅读本文)

一、评测环境

评测在 64 位 windows 10 系统下使用 Project Lemon Plus 进行。C++ 编译器为 g++, 版本号 4.9.2。

评测时, 程序可用的栈空间大小与题目空间限制一致。也即你可以自由地使用系统栈而无需担心爆栈问题。需要指出的是, 系统所使用的动态内存空间 (包括系统栈空间, 系统堆空间) 会被计入程序所使用的空间大小。程序运行所占用的总空间为动态空间的峰值与静态空间之和, 其大小不能超过题目空间限制, 否则将会被判定为 MLE。另外需要说明的是, 这里的静态空间是指程序声明的静态空间而不是实际使用的静态空间。

评测机的硬件参数为 Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz, 内存 16GB。其效率略高于 NOIp 新评测机, 但差别不大。

为了让你直观的了解评测机的效率, 经过测试, 读入 10^6 个 long long 范围内的数字并插入 std::set 中所需要的时间为 800ms。测试所用的程序在选手文件夹下的 test\testcpp.cpp, 所插入的数据在选手文件夹下的 test\testdata.in。你可以直接在该文件夹下编译运行该 cpp 文件, 通过观察标准错误流 (会反馈到控制台) 的输出来获得该程序在你的电脑上所运行的时间, 据此判断选手机器与评测机的效率差异。测试时的编译开关为 -O2, -std=c++11。

64 位 Windows 10 对于 64 位有符号整型的占位符为 %lld, 对于 64 位无符号整型的占位符为 %llu。如果你希望使用 printf/scanf 来输出/读入一个 long long/unsigned long long, 请使用上述占位符。使用 %I64d/%I64u 将被视为 UB。

可以使用万能头文件, 但是不建议这么做。

二、评测方式

评测方式分为两种: 按测试点评测和捆绑测试。

按测试点评测即为传统评测方式。通过一个测试点即可得到对应分数。

对于捆绑测试的题目, 测试时会有若干个子任务, 每个子任务有一定的分数, 下辖若干个测试点。对于一个子任务, 只有通过其下所有测试点, 才能得到该子任务的分数, 否则该子任务得 0 分。

三、输入文件

输入文件在 windows 下以 wb 参数生成。行末有且仅有一个 \n 换行符, 没有 \r 换行符。你无需担心换行符对你的程序造成影响, 可以正常进行读入。

对于一行输入多个数的情况, 各个数之间均以单个空格隔开, 行末无多余空格。

保证输入的末尾有一个换行, 但是没有多余空行。

四、附加样例文件

附加样例文件存放在题目目录下。其生成方式与输入文件相同。

五、源代码存放及文件夹命名规定

对于每道题目, 请建立一个名为题目名的子文件夹。例如, 题目 forever 的源代码应该存放为 forever/forever.cpp。为了避免文件夹大小过大, 在提交前请尽量删除文件夹内的可执行文件 (exe) 和过大的输入/输出文件, 但不做强制要求。

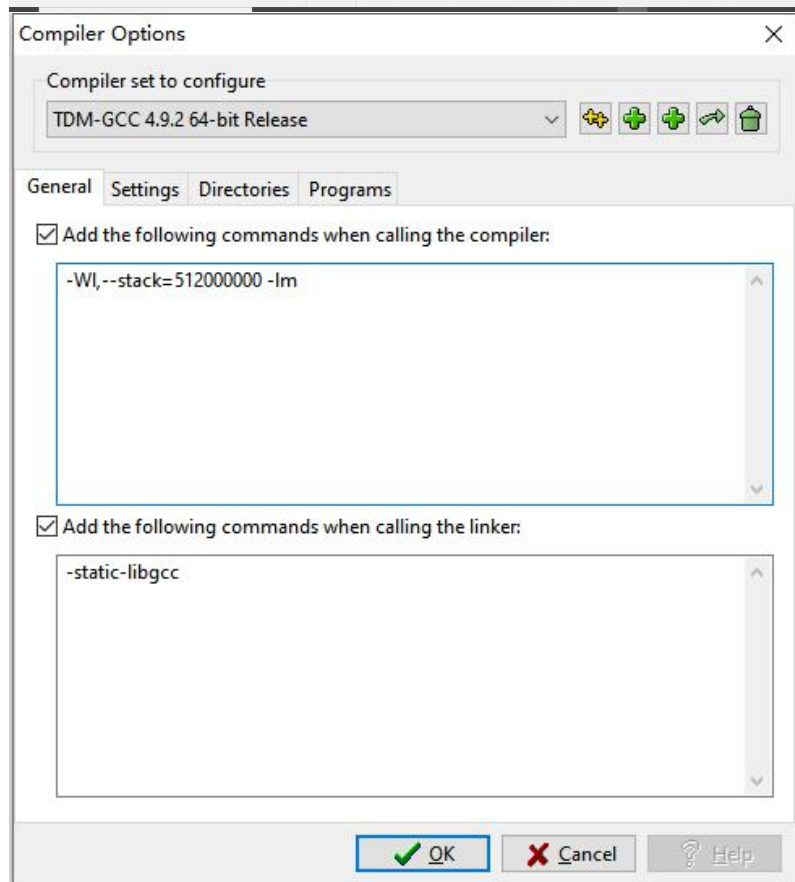
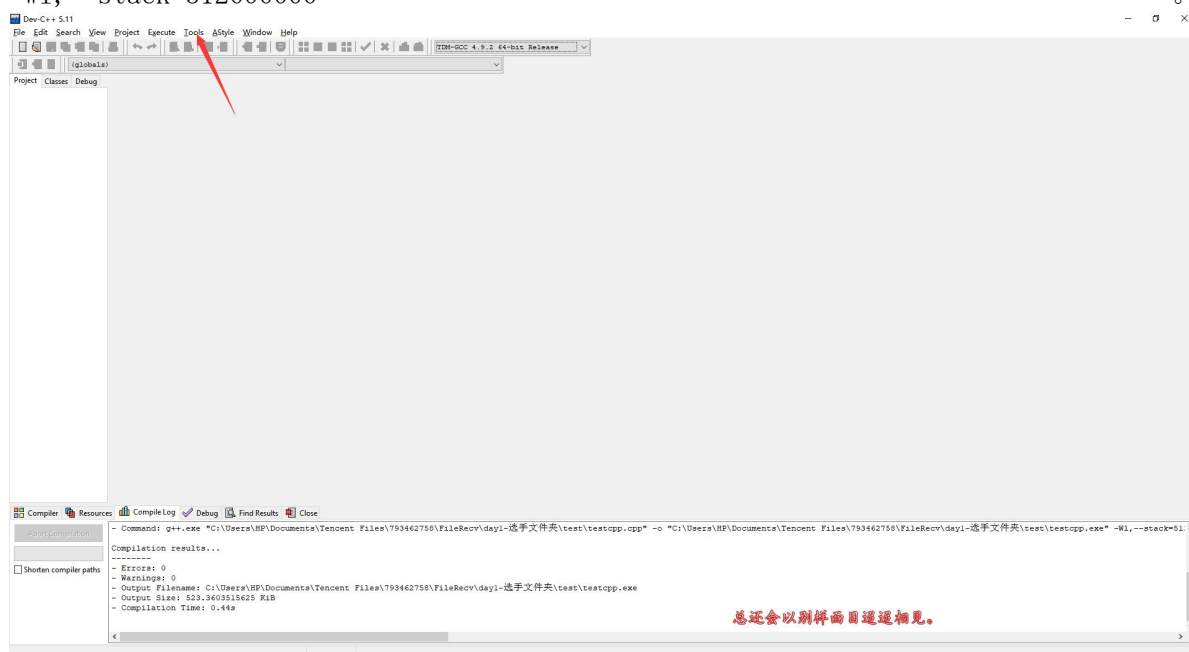
为便于统计成绩, 选手文件夹两天命名应该相同。

竞赛结束后, 曹宇琮负则收齐代码并提交。

六、编译参数

编译时不开启任何优化开关。在测试附加样例文件时，由于数据过大在 windows 下可能出现爆栈的情况。在本机开启无限栈的方式为：

在「工具」一栏下找到「编译选项」选项卡，然后勾选第一个选项，并在框中输入 `-Wl,--stack=512000000`



七、申诉规定

只接受以卡常为原因的申诉。其他原因一概不予重测。