好的，组长同学。你们的情况非常典型（纯C基础、无项目经验、时间紧），但完全有可能成功。作为组长，你的角色至关重要。

我为你和你的小组制定了一个为期四周的详细作战计划，并指出了合作中的关键点。

整体战略 (Overall Strategy)

1. 目标分解 (Divide & Conquer)：将庞大的IDE项目分解成多个小模块，小组并行开发。

2. 增量开发 (Incremental Development)：每周都有一个可运行、可演示的版本。不要想着一口吃成胖子。

3. 优先核心 (Core First)：优先实现最基本的功能（编辑+编译运行），再考虑美化界面和添加高级功能。

4. 持续沟通 (Constant Communication)：每天同步进度，快速解决问题，避免有人掉队或方向走偏。

---

四大阶段详细规划 (4-Week Detailed Plan)

总前提： 本周内（9月3日前）所有人必须完成开发环境搭建：安装 Qt Creator 和 MinGW-w64（或你用的C编译器）。

---

🏗️ 第一阶段：奠基与探索 (9月2日 - 9月8日) - 全员攻坚

目标： 搭建程序框架，实现一个最最最简单的文本编辑器（能打开、保存、编辑文字），并搞明白如何调用编译器。

· 任务 1 (所有人)：Qt 紧急入门 (2天)

· 统一观看同一个Qt入门教程（例如“Qt Creator快速入门”系列视频），学习核心概念：QWidget, QMainWindow, QMenuBar, QStatusBar, 最重要的是 QPlainTextEdit (核心的文本编辑组件)。

· 每人至少跟着教程做一个带菜单的空白窗口程序。

· 任务 2 (2人小组： “编辑器组”)：核心编辑器模块

· 使用 QPlainTextEdit 组件。

· 实现：新建、打开、保存、另存为 功能。（使用 QFile 和 QTextStream）

· 输出： 一个可以独立运行的简易记事本。

· 任务 3 (2人小组： “编译器组”)：核心编译运行模块

· 研究如何使用 QProcess 类。

· 目标：写一个函数，能执行命令 gcc -o output.exe input.c，并能捕获命令行的输出（编译错误信息）和返回值。

· 输出： 一个独立的控制台程序，能输入C文件路径，调用gcc编译并打印结果。

· 任务 4 (组长 + 1人)：项目框架与集成 (后3天)

· 组长创建主项目，搭建主界面（菜单栏、中央文本编辑器、底部的状态栏、侧边的输出窗口）。

· 将“编辑器组”的代码整合进来。

· 将“编译器组”的 QProcess 调用代码整合进来，添加 编译 和 运行 菜单项。

· 本周里程碑 (Milestone):

· 周末 (9月8日) 必须产生第一个可运行版本：

· 可以编辑代码。

· 点击“编译”，能在下面的输出窗口看到编译成功或失败的错误信息。

· 点击“运行”，能弹出黑色控制台窗口显示程序输出。

---

⚙️ 第二阶段：功能实现与集成 (9月9日 - 9月15日) - 深度开发

目标： 完善基本功能，实现可选功能中最实用的部分。

· 任务 1 (2人)：增强编辑器功能

· 实现：查找、替换、复制、粘贴、剪切。（QPlainTextEdit 自带部分功能，需要将其连接到菜单项）

· (可选) 研究语法高亮（这是一个难点，可以参考 QSyntaxHighlighter）。

· 任务 2 (2人)：增强编译运行与UI

· 实现：编译时显示“编译中...”，完成后显示“成功”或“失败”。

· 设计运行程序的输出窗口（可以用另一个 QPlainTextEdit 来显示程序输出的内容，而不是弹出黑框）。

· 添加“停止运行”按钮（终止 QProcess）。

· 任务 3 (1人)：研究与实现额外小功能

· 例如：行号显示（QPlainTextEdit 周边放一个 QWidget 来画行号）、括号匹配高亮。

· 本周里程碑 (Milestone):

· 9月15日开学前，得到一个功能比较完善的IDE原型。 基本编辑功能和编译运行流程必须全部畅通。

---

🐞 第三阶段：测试与优化 (9月16日 - 9月22日) - 精益求精

目标： 疯狂测试，修复Bug，提升稳定性，完善UI/UX。

· 全员压力测试：

· 每个人用自己的IDE写一些C程序（几十行到几百行），进行编译、运行。

· 专门找茬：尝试各种非法操作（比如编译时疯狂点击运行、打开非C文件、删除正在编辑的文件等），确保程序不会崩溃，而有友好的提示。

· 记录下所有遇到的Bug。

· 成立“Bug修复小组”(2人)： 专门负责解决大家提交的Bug。

· 其他人： 继续实现1-2个最吸引人的可选功能，如：

· 代码折叠（较难）

· 简单的自动补全（更难）

· 程序自动排版（调用 astyle 等外部工具）

· 首选推荐：主题切换（浅色/深色模式），这个相对简单且演示效果出色。

---

📦 第四阶段：交付准备 (9月23日 - 9月29日) - 完美收尾

目标： 整理代码、撰写文档、录制演示视频。

· 代码整理： 统一代码风格，写注释，删除调试用的无用代码。

· 文档撰写 (分工完成)：

· 《需求分析文档》：说明我们实现了什么。

· 《软件设计文档》：画一下简单的类图，说明用了哪些Qt类。

· 《开发管理文档》：最重要！ 详细记录每个人的分工（用表格形式）。

· 《个人总结》：每人写自己的心得体会。

· 录制演示视频 (1-2人)： 提前写好脚本，展示最完整、最流畅的操作流程（3-5分钟为宜）。

· 最终打包提交。

---

👥 合作难点与要点 (作为组长的你务必注意)

1. 版本控制 (Git) 是生命线！

· 难点： 你们可能没人用过Git，代码合并会是一场灾难。

· 解决方案：

· 立即在Gitee或GitHub上创建私有仓库。

· 本周内，全员必须学会最基础的Git操作：clone, pull, add, commit, push。

· 规定： 每天工作结束前，必须将当天代码push到仓库。每天开始工作前，第一件事是pull更新代码。

· 组长负责处理合并冲突。

2. 明确的接口约定

· 在开始各模块开发前，小组长要组织会议，明确约定模块之间的接口。

· 例如： “编译器组”需要提供一个类似 bool compileFile(const QString &filePath) 的函数，返回是否成功，错误信息存在一个全局字符串里。这样“编译器组”和“界面组”就可以并行开发，而不用等对方的代码。

3. 每日站会 (Daily Stand-up)

· 每天固定时间（比如晚上9点），开15分钟腾讯会议。

· 每人轮流说三件事：① 昨天做了什么？② 今天计划做什么？③ 遇到了什么困难？

· 这是防止有人掉队、发现问题的最有效手段！

4. 保持简单，避免“炫技”

· 时刻提醒组员，我们的目标是按时完成一个稳定可用的核心功能，而不是做一个比Visual Studio还强大的IDE。对于那些难度过高、耗时过长的可选功能（比如Debugger），要敢于放弃。

5. 心态管理：鼓励与互助

· 有人卡住了，鼓励其他同学一起去帮他查资料、看代码。

· 多肯定大家的进度，庆祝每一个小里程碑的达成（比如第一次编译成功时，可以在群里发个红包）。

组长，你的主要职责不是写最多代码，而是：

· 规划者 (Planner): 把握大方向，分解任务。

· 协调者 (Coordinator): 组织会议，协调分工，解决冲突。

· 督促者 (Supervisor): 跟踪进度，确保没人躺平。

· 兜底者 (The Last Defender): 在最后关头，哪个模块最危险，你就去支援哪里。

时间非常紧张，但只要计划得当、沟通充分，你们一定能成功完成这个挑战。这不仅是一次编程作业，更是一次宝贵的团队合作实战经验。加油！