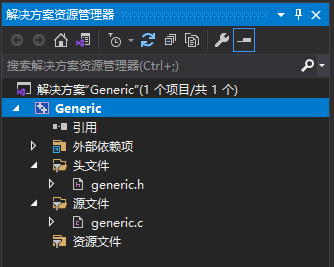
1）准备Windows环境，并安装Visual Studio

如果你使用Mac，请通过虚拟机安装Windows。

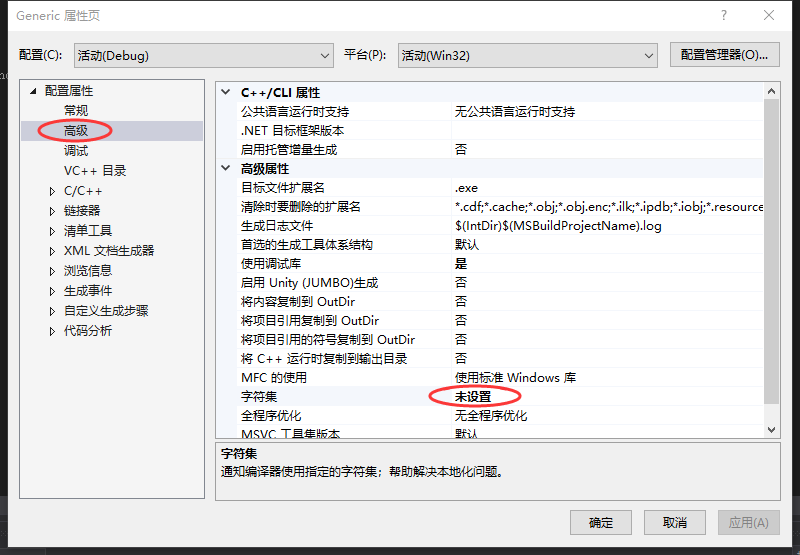
下载Visual Studio时选用Community版本（免费），安装时确保C++语言被勾选。

安装完成后，打开VS，新建项目，C++，Windows桌面应用程序，创建。

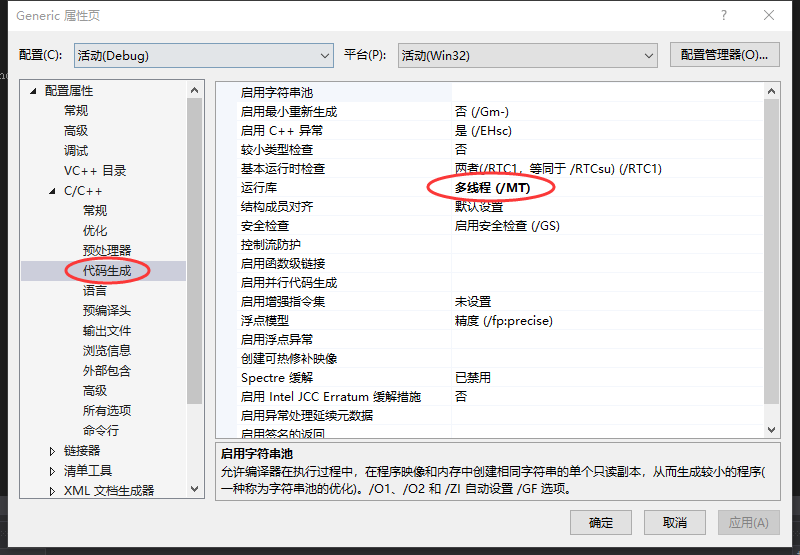
然后，把老师提供的文件复制到项目文件夹，并在VS的解决方案资源管理器中，删除所有自动生成的文件（包括资源文件），插入老师提供的文件。



还是在解决方案资源管理器中，右击项目名，编辑项目属性，高级，字符集，设置为“未设置”。



接下来，尝试运行，如果报错，项目属性，C/C++，代码生成，运行库，设置为“多线程（/MT)”，再运行。



成功运行后，尝试点击窗口空白处，看到消息框。

修改程序，使消息框中增加你的学号、姓名，提交运行截图。

2）准备Linux环境，并配置C++开发环境

**如果你使用Mac：**

我不了解Mac，据我所知Mac对本课程用到的库函数都兼容，所以应该不用另外安装一个Linux虚拟机。如果将来作业发现问题，再安装虚拟机不迟。C++开发环境自行配置。完成后，编译老师提供的程序，发现错误，纠正，并在输出结果中增加你的学号、姓名，编译运行，提交截图。

**如果你使用Windows，有两种选择：**

选择一：传统方式（如VMWare）安装虚拟机

自行安装虚拟机软件（如VMWare），下载Linux镜像文件（如Ubuntu 22.04），安装即可。

操作系统安装完成后，先执行sudo apt-get update命令，把“应用市场”的软件列表获取到本机。

通过sudo apt-get install build-essential gdb命令下载安装C++编译器和调试器。

接下来安装并配置自己熟悉的C++开发环境。例如，VSCode如下：

通过sudo apt-get install code下载安装VSCode（也可以Web下载VSCode的.deb安装包，再用dpkg命令安装）。

把老师提供的源程序复制到“/home/用户名/”下的某个文件夹，作为项目文件夹。

启动VSCode，下载“C/C++”插件，然后打开项目文件夹，接下来为该项目配置代码提示、编译、调试命令，即配置c\_cpp\_properties.json、launch.json、tasks.json，配置方法上网搜。

编译老师提供的程序，发现错误，纠正，并在输出结果中增加你的学号、姓名，编译运行，提交截图。

优点：Linux系统完整，边界清晰，初学者知道自己在哪里，在做什么，有图形界面。

缺点：占用硬盘、内存多，系统比较封闭，跟Windows之间传输文件较麻烦，可能还需要给Linux安装个SSH Server，sudo apt-get install openssh-server，提供SSH远程登录，然后在Windows端安装个FileZilla客户端，通过SFTP协议登录Linux系统，上传下载文件。

选择二：利用Windows自带的WSL（Windows Subsystem for Linux）安装虚拟机

按照老师提供的文档，给Windows安装WSL组件，然后安装Linux系统。再配置开发环境，在Linux端安装C++编译器和调试器，在Windows端配置VSCode（你也可以选用配置CLion或其它IDE）使其远程登录到Linux系统执行开发工作。

编译老师提供的程序，发现错误，纠正，并在输出结果中增加你的学号、姓名，编译运行，提交截图。

优点：Linux端默认不提供图形界面，占用硬盘、内存少，与Windows之间互访文件方便（所以没有图形界面影响不大），开发期间基本停留在Windows端，避免两套桌面来回切换。

缺点：Linux端没有封闭的界面，边界感不强，开发时须借用远程登录（只有VSCode这方面做得比较好），集成开发环境在Windows这边，编译器调试器在Linux这边，关系略复杂。