阅读指南(请认真仔细阅读)：

1、大作业的统一说明及要求

2、下发的大作业模板的使用说明

3、在多项目中添加公共函数的基本方法

4、公共函数开发及维护的基本准则

5、大作业的命名要求及存放目录要求

6、大作业的提交要求

7、大作业的评分标准及注意事项

8、后语

1、大作业的统一说明及要求

★ 本学期会有若干大作业，每个作业的完成时间为1~3周

★ 后面的大作业，可能会跟前面相似(例如：界面部分)，也可能是对前面大作业提出新要求

★ 为了培养多源程序划分、提炼公共函数、多项目共用公用函数的工程素养，下发大作业的VS2022解决方案模板，请认真阅读命名要求、提交要求并应用于后续所有大作业的完成与提交中

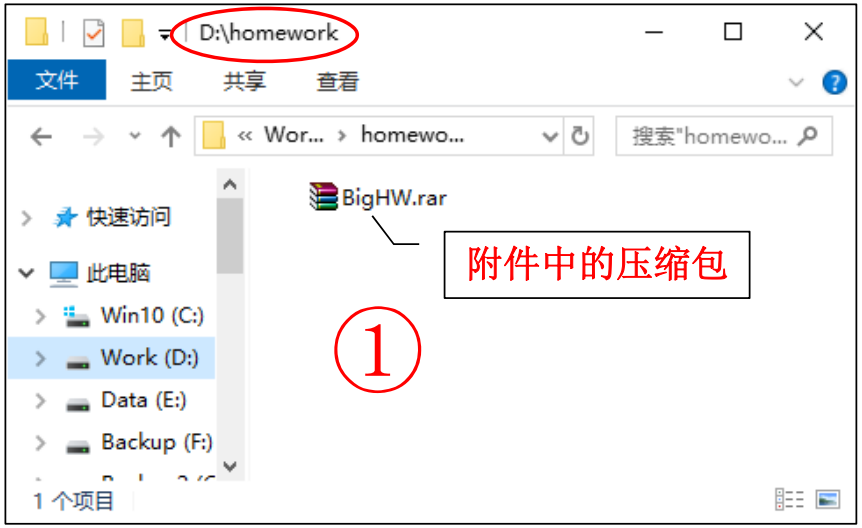
2、下发的大作业模板的使用说明

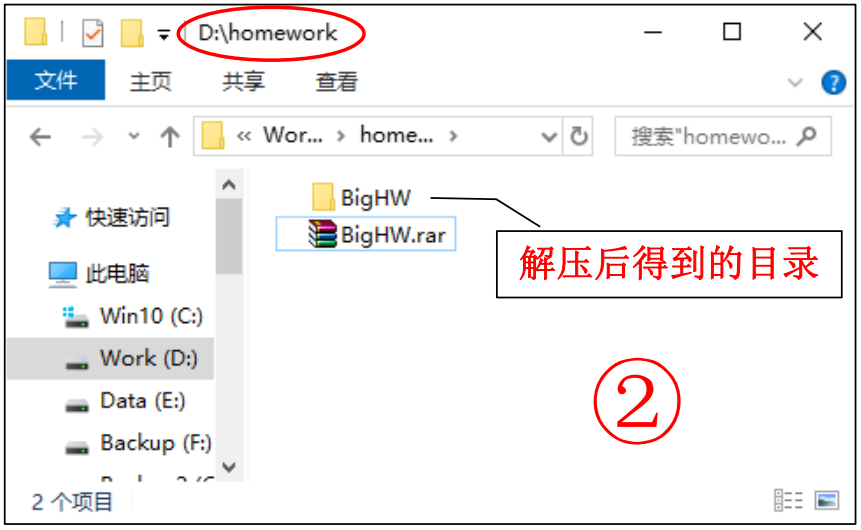
Step-1：选择合适的盘符/目录，将附件BigHW.rar放在此目录下并解压，得到 BigHW 目录

本文档以"D:\homework"为主目录为例进行说明(实际操作时可以按需定义自己的主目录)

① 在D盘建立homework目录，将附件中的BigHW.rar文件放入

② 解压时选择"解压到当前文件夹"即可(不同解压软件可能说法不同)



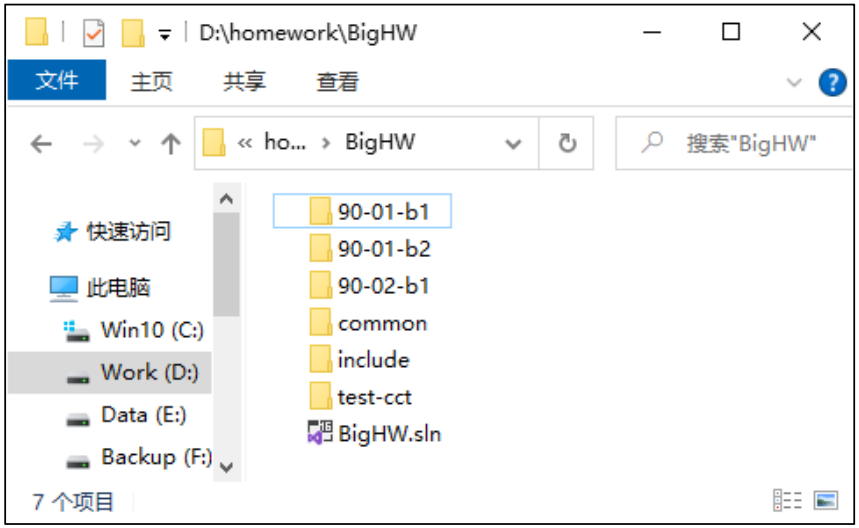


2、下发的大作业模板的使用说明

Step-2：进入 BigHW 目录，能看到大作业的解决方案文件 BigHW.sln 及6个子目录

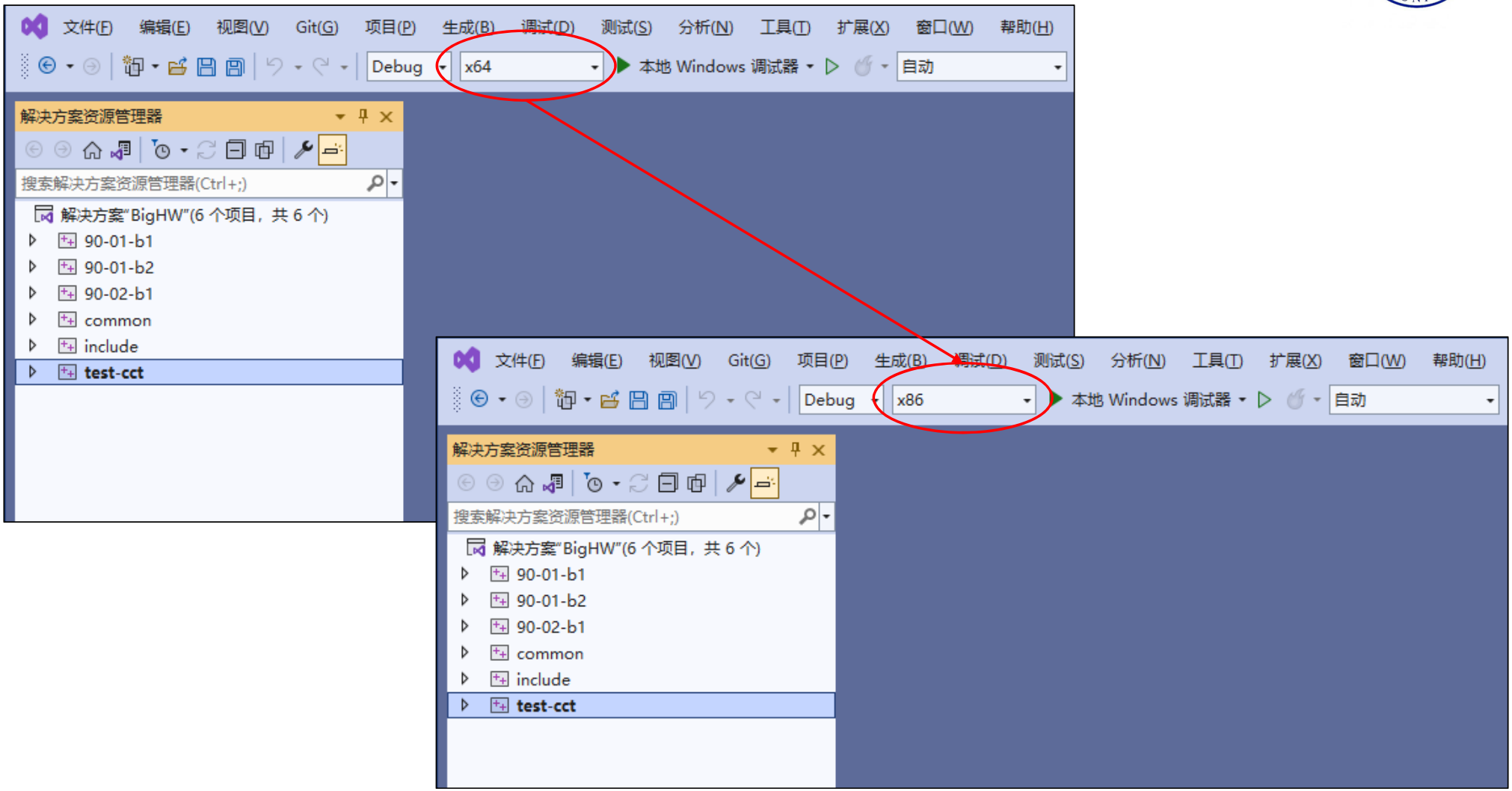
★ 目前的6个目录的存放内容约定如下

- include ：存放所有大作业都会用到的头文件(存放宏定义/公共函数声明等)
  - ◆ 目前仅有 cmd_console_tools.h 一个公共函数头文件，后续陆续添加
- common ：存放所有大作业都会用到的源程序文件(存放公共函数的实现)
  - ◆ 目前仅有 cmd_console_tools.cpp 一个公共函数源文件，后续陆续添加
- test-cct ：存放给出的cmd_console_tools中系列函数的测试用例
  - ◆ 目前仅有 test-cct.cpp 一个源文件(注：认真阅读文件开始的头文件包含说明)
- 90-01-b1 ：存放上学期的第一个大作业（汉诺塔）的所有源文件及头文件(具体命名见作业要求)
  - ◆ 目前仅有90-01-b1-main.cpp且内容为空，具体见作业要求
- 90-01-b2 ：存放上学期的第二个大作业（合成十）的所有源文件及头文件(具体命名见作业要求)
  - ◆ 目前仅有90-01-b2-main.cpp且内容为空，具体见作业要求
- 90-02-b1 ：存放本学期的第一个大作业的所有源文件及头文件(具体命名见作业要求)
  - ◆ 目前仅有90-01-b1-main.cpp且内容为空，具体见作业要求
- 90-02-bx ：存放本学期的后续大作业（待追加）

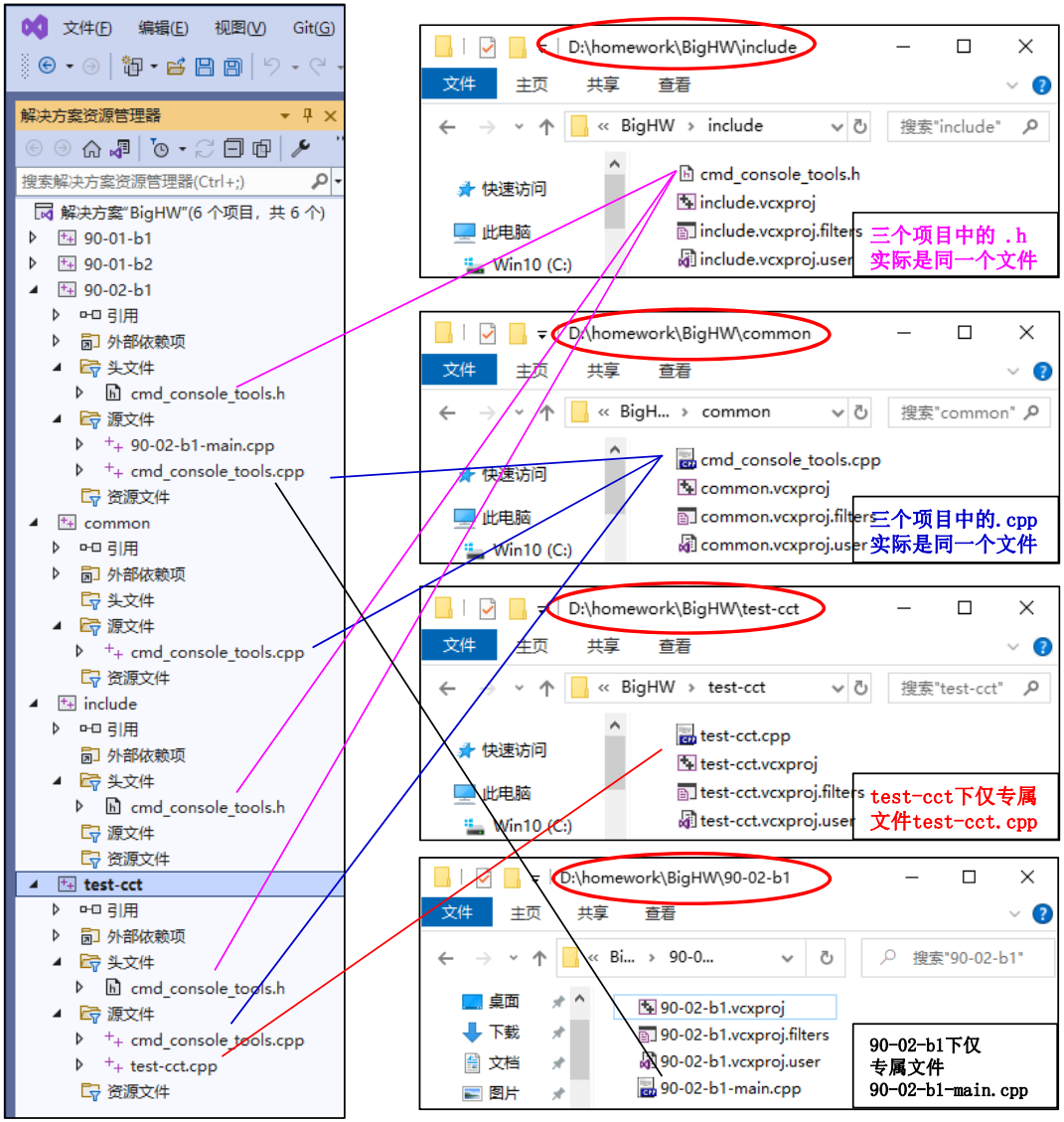


2、下发的大作业模板的使用说明

Step-3：双击 BigHW.sln，打开解决方案，将配置管理器中缺省的 x64 改为 x86 !!!



Step-4：解决方案中各项目中公用文件的说明



说明：

1、如解决方案资源管理器所示

- include/test-cct/90-02-b1 这三个项目中均有文件 cmd_console_tools.h
- common/test-cct/90-02-b1 这三个项目中均有文件 cmd_console_tools.cpp

2、如各项目对应的文件目录所示

- cmd_console_tools.h 只存在于 D:\homework\BigHW\include中
- cmd_console_tools.cpp 只存在于 D:\homework\BigHW\common中
- D:\homework\BigHW\test-cct 中只有专属文件 test-cct.cpp
- D:\homework\BigHW\90-02-b1 中只有专属文件 90-02-b1-main.cpp

3、这种项目结构下，在任意一个项目中修改公共文件，对应的是同一个文件，若其他项目需要再次编译时，同样会编译被修改过的公共文件

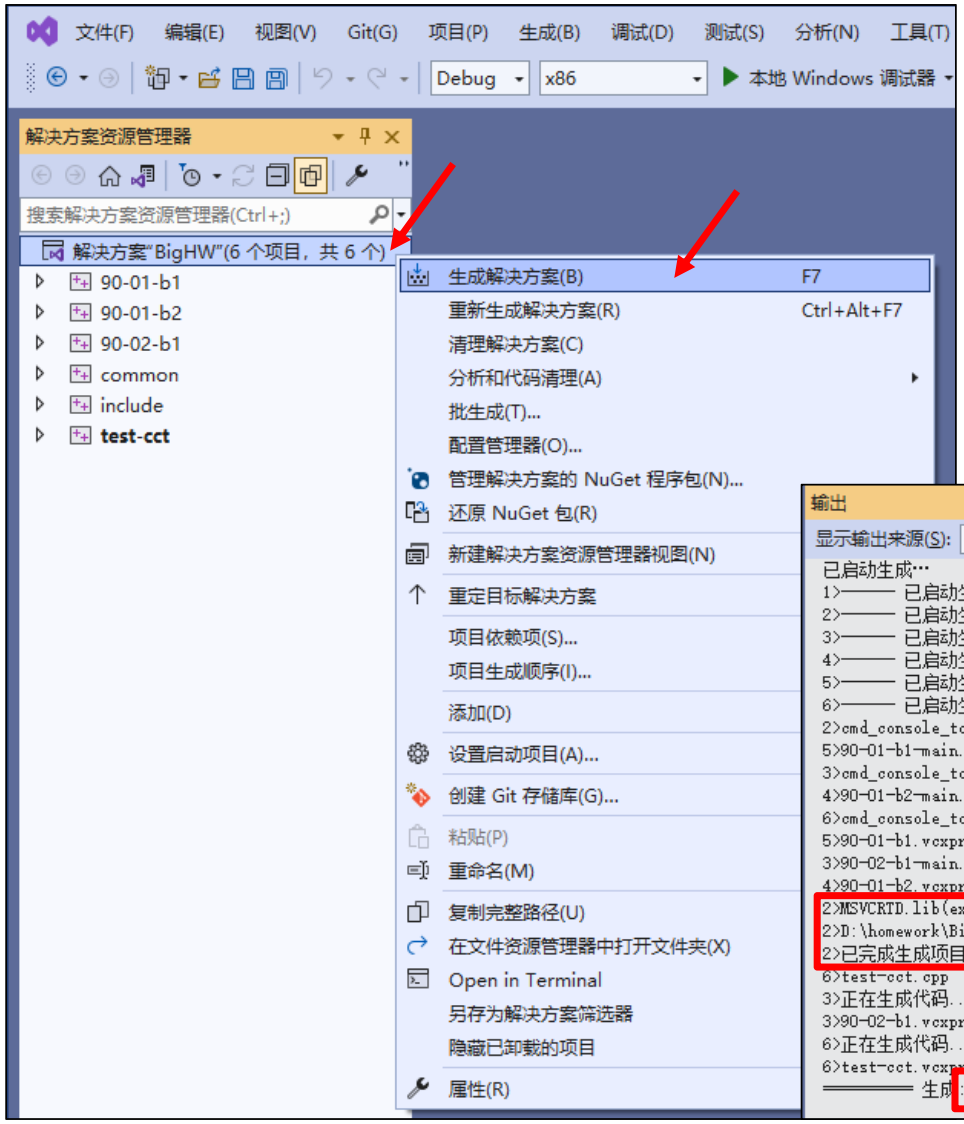
（例：在test-cct项目中编辑源文件 cmd_console_tools.cpp，其它两个项目中对应的同名cpp文件会同步修改，若再次编译90-02-b1，则会重新编译修改后的.cpp）

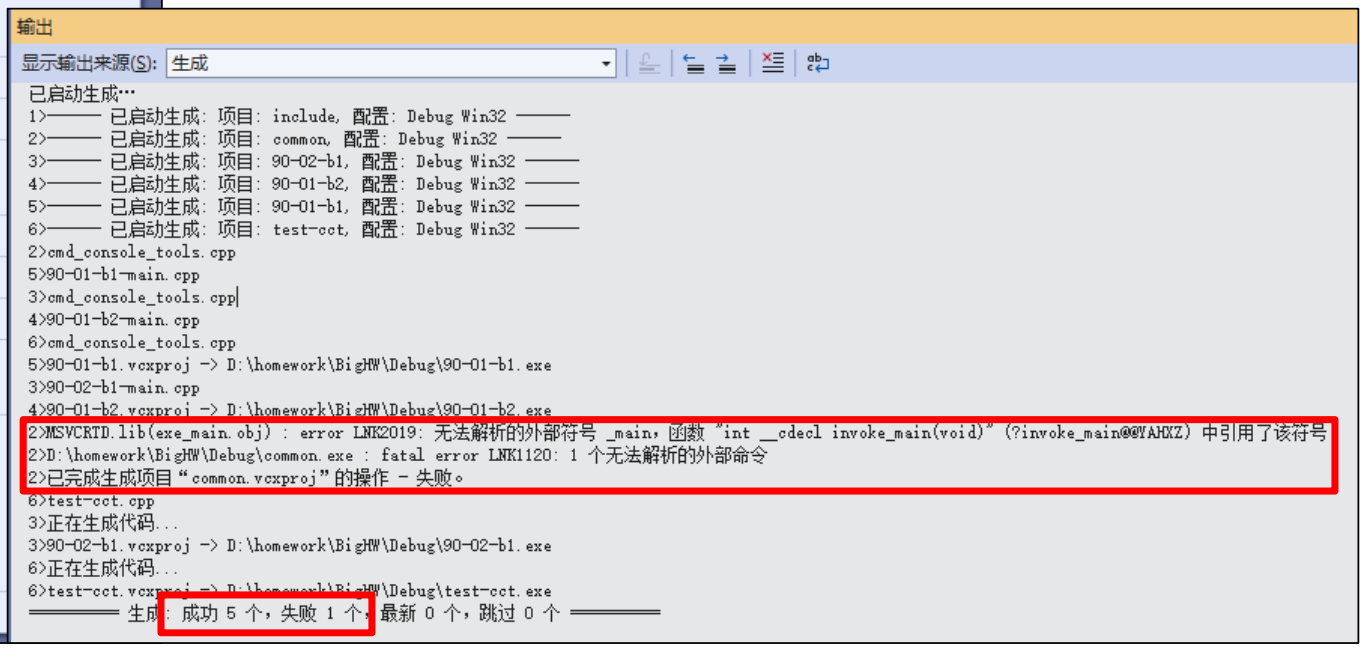
4、公共文件使用总结：

- 公共文件是指仅在一个目录下存在，但是被多个项目引用的文件
- 优点：在任意项目中对公共文件的修改都可以使多个项目同步修改，使用方便，不必维护多份拷贝
- 缺点：修改要谨慎，要特别注意公共文件的多项目适应性，避免以前编译通过的项目再次编译会错误!!!
- 在项目中添加其它目录中文件的方法见后

【注：】本例中，不要试图将common或include项目设置为启动项目，因为两个项目都不完整，编译会出错，放在这里的目的只是为了查看方便，且整体生成时，会排除这两个项目

附加：如何在解决方案中排除common/include项目





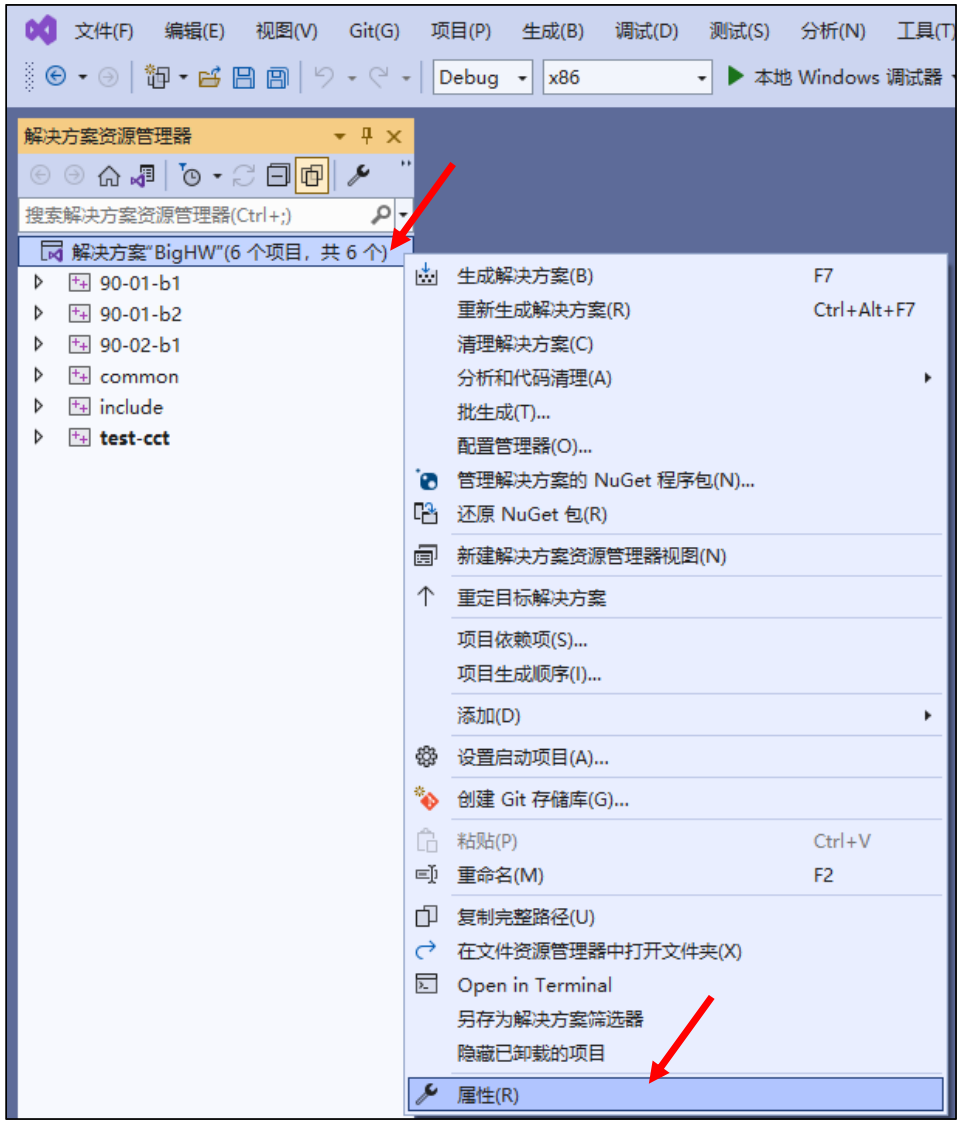
如果所示6个项目，如果在“解决方案”上右键-“生成解决方案”
- （即生成该解决方案的全部项目的exe）

则common项目会报错（因为这不是一个完整的项目，无main函数）

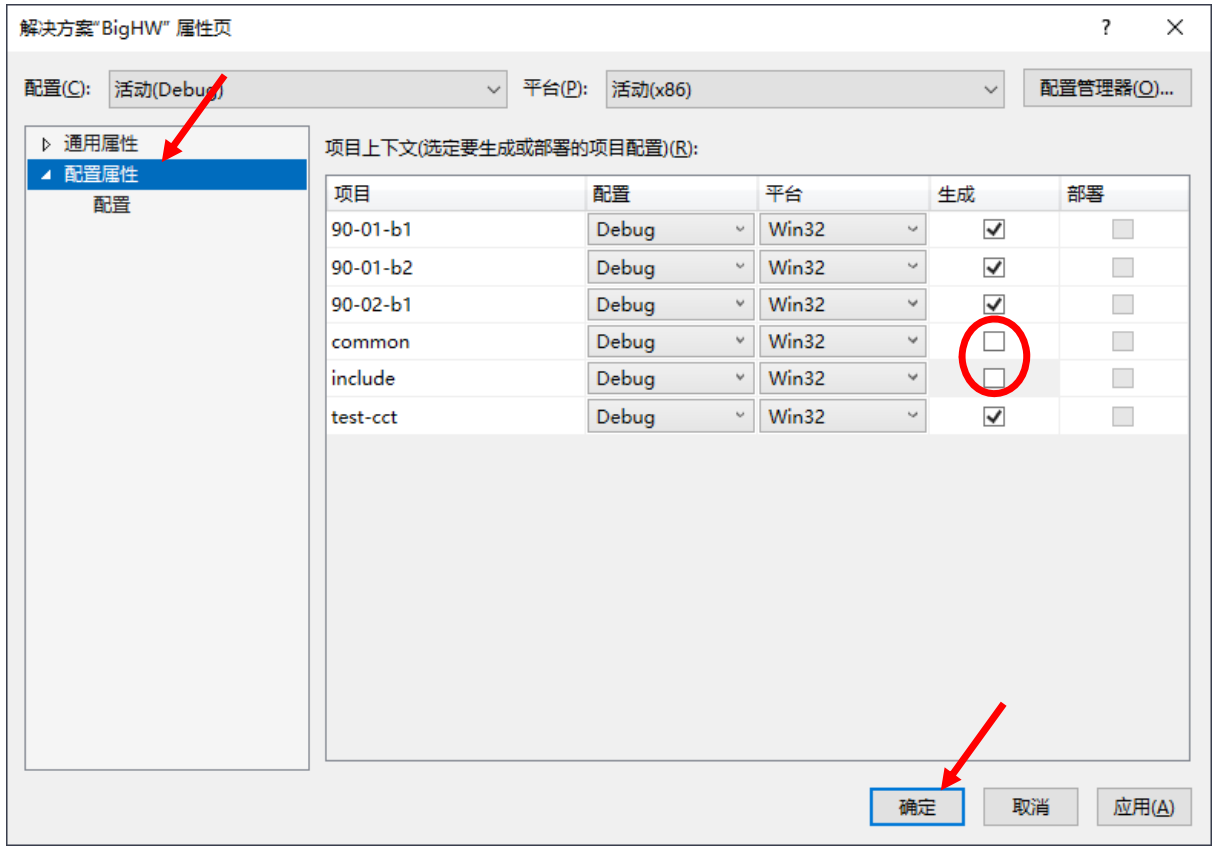
注：include项目不报错是因为里面没有cpp文件

因此在生成解决方案时，需要排除common和include

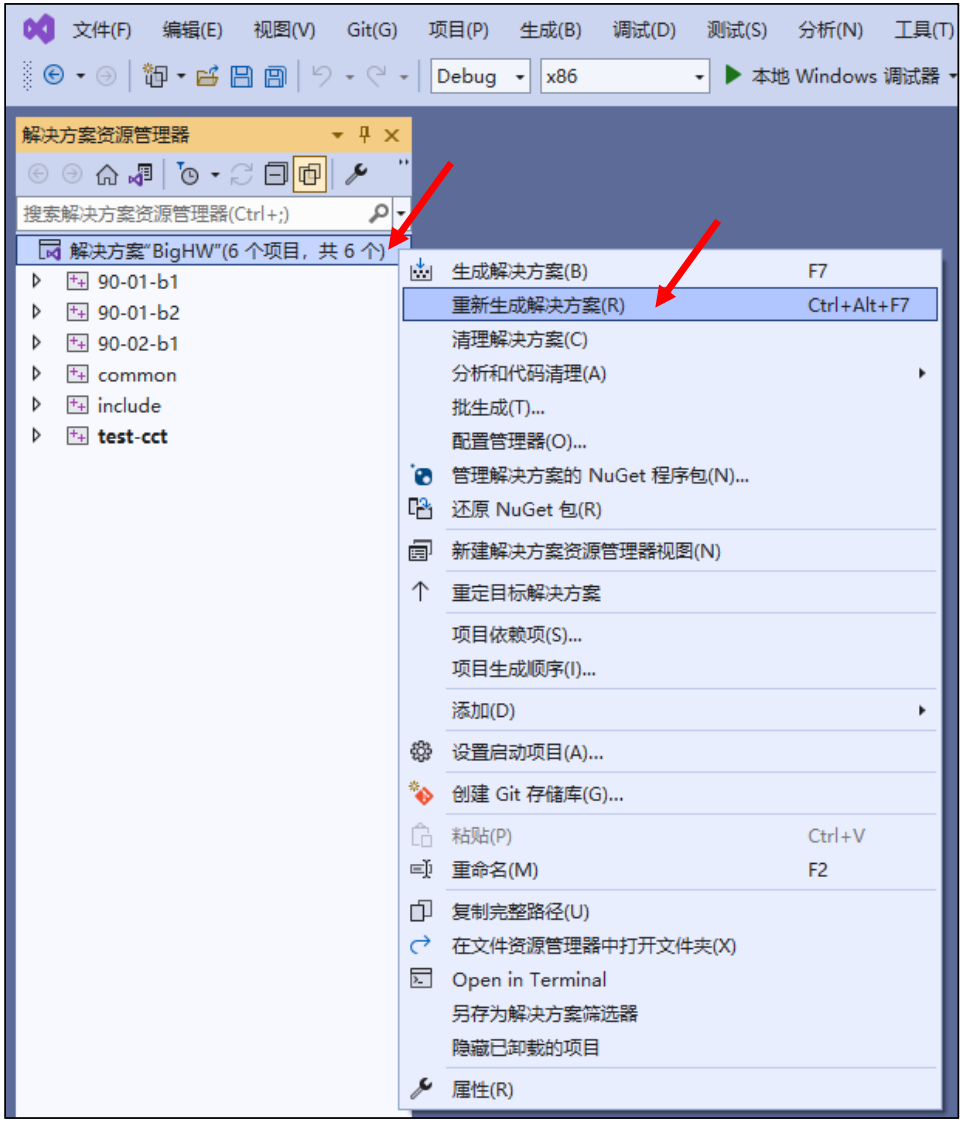
附加：如何在解决方案中排除common/include项目



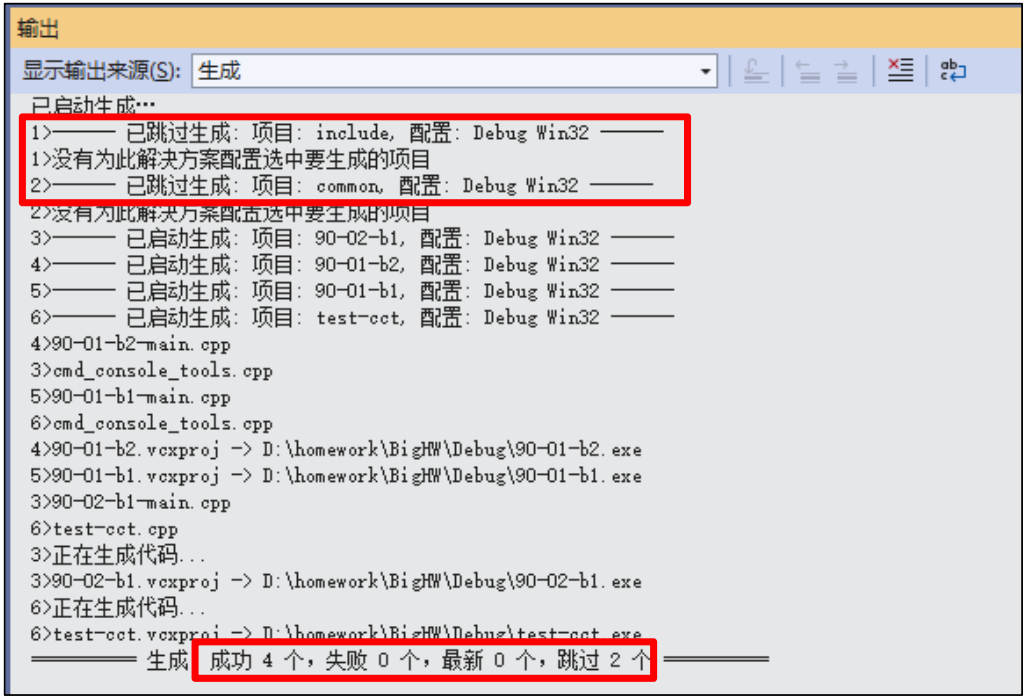
在“解决方案”上右键-“属性”
出现的属性页设置中，选“配置属性”，去除common/include在Debug/Win32下的生成选项，按“确定”即可



附加：如何在解决方案中排除common/include项目

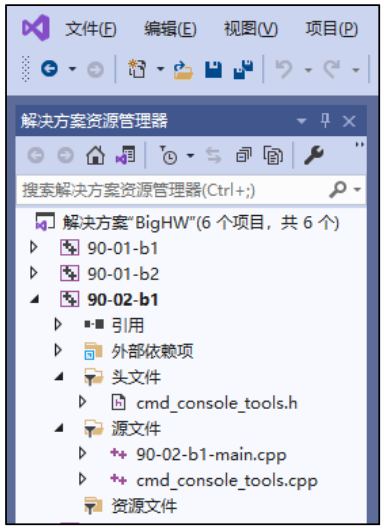


在“解决方案”上右键-“重新生成解决方案”
- 也可以先“清理解决方案”，再“生成解决方案”
则common/include项目会跳过



3、在多项目中添加公共函数的基本方法

3.1、在新的项目中添加已有公共函数（以下发模板中90-02-b1项目的建立为例）：



基本步骤：

1、鼠标右键单击解决方案"BigHW"，菜单中"添加"-"新建项目"，添加新项目90-02-b1

2、鼠标右键单击项目"90-02-b1"，菜单中"添加"-"新建项"，选择"C++文件"，
加入90-01-b1-main.cpp

3、鼠标右键单击项目"90-01-b1"，菜单中"添加"-"现有项"，出现的对话框中移动到
"BigHW"目录，再进入"include"目录，选择cmd_console_tools.h后"添加"即可

4、同理加入common目录下的cmd_console_tools.cpp

3.2、新建公共函数并添加到已有项目中

基本步骤（以cmd_input.h/cmd_input.cpp为例）：

1、在"include"项目中添加需要的头文件cmd_input.h

2、在"common"项目中添加需要的源程序文件cmd_input.cpp

★ 建议头文件/源文件的主文件名相同（例：cmd_input）

3、在90-01-b1/90-01/b2/90-02-b1等项目中添加 cmd_input.h / cmd_input.cpp （步骤参考3.1）

★ 此步骤一定要做，即公共函数增加后，一定要在各需要的项目中添加

★ 再次强调：不要把 cmd_input.h、cmd_input.cpp 文件复制到各项目目录中!!!

4、在cpp中包含其它目录下头文件的方法请认真阅读附件中的代码，里面有详细说明

4、公共函数开发及维护的基本准则

1、维护一套通用的公共代码对于程序的开发是极其重要的，能大大减轻后续开发的工作量，并减少相同/相似代码多处复制带来的不统一性

2、公共代码改动后，之前所有代码都需要再次测试，这个步骤不可以省略!!!

3、公共代码的后续改动导致前面已通过测试的代码出问题是不可接受的!!!

4、做好备份，培养自己维护好历史版本的能力(可以自动定义一套规则，也可尝试借助一些工具)

★ VS自带Git，尝试去了解Git、申请Git账号并掌握基本操作（不是本课程要求，但做为计算机人，这是基本要求）

5、大作业的命名要求及存放目录要求

★ 每个大作业单独用的文件，放于每个项目子目录下

（例：完成90-02-b1作业用的私有源程序文件/头文件，放在90-02-b1子目录下）

★ 每个大作业单独用的文件，命名为 90-0x-bx-***.cpp/.c/.h，具体命名及文件个数自行定义

（例：90-02-b1的私有文件，命名为 90-02-b1.h / 90-02-b1-main.cpp / 90-02-b1-base.cpp / 90-02-b1-tools.cpp 等）

★ 存放两个或两个以上项目公用函数的头文件，必须放于include目录下；

存放两个或两个以上项目公用函数的源程序文件，必须放于common目录下；

这两个公共目录下的文件，命名为 common_-***.cpp/.c/.h，具体命名及文件个数自行定义

（例：假设90-01-b1/90-01-b2都要用到几个从键盘输入不同类型数据的函数，则可以把这些公共函数的声明放在include下的common_input.h中；这些公共函数具体实现的源程序文件则放在common下的common_input.cpp中）

★ include/common下模板中已给出的cmd_console_tools.h/cmd_console_tools.cpp，不准改名，不准修改内容

★ 每个头文件及源程序文件均有首行要求!!!(违规会扣分)

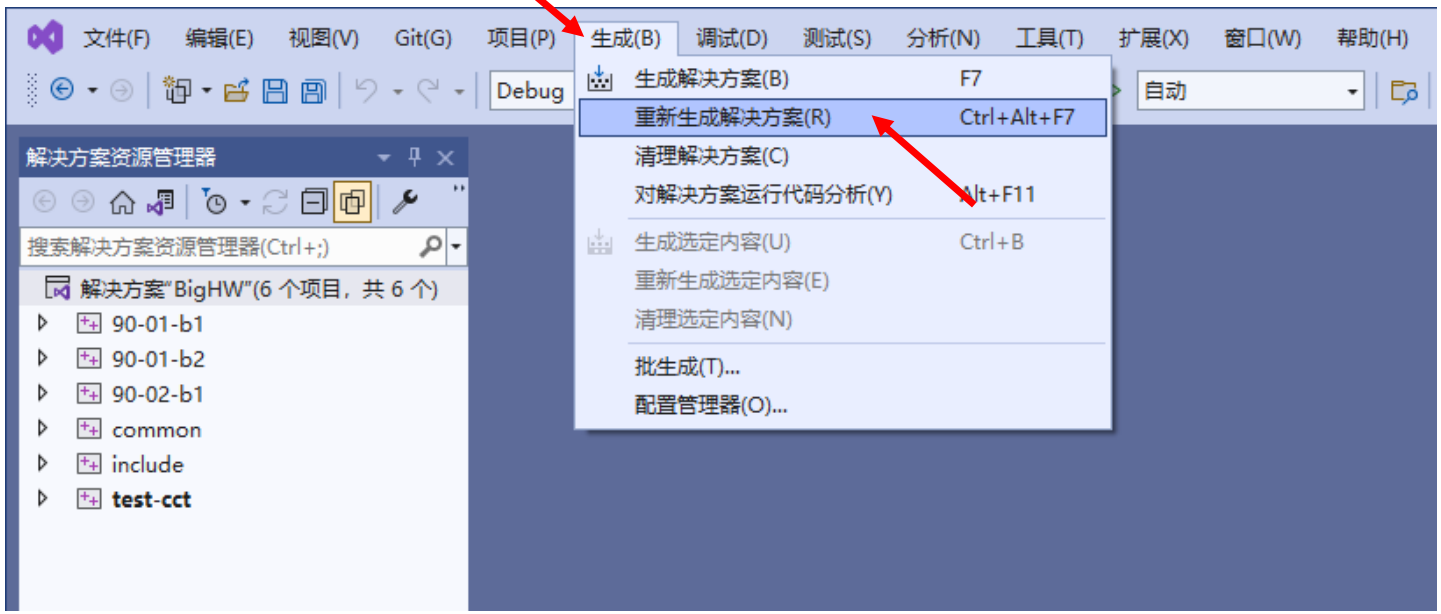
★ 建议：include/common下所有的公共函数的声明及实现，建议自定义一套函数命名规则，以使自己编写程序时能很方便地知道某函数的功能及适用范围

★ 建议：每次修改公共函数后，最好把之前已编译通过的项目再验证一遍(若出错会扣分)

（例：第一次大作业90-02-b1完成后，编译通过；继续完成第二次大作业90-02-b2，编译通过，但是第二次大作业提交后，发现90-02-b1编译错误，则会扣除90-02-b1之前的得分）

6、大作业的提交要求

Step-1：完成所有测试后，先选择"清理解决方法"，再关闭 VS2022（或关闭BigHW解决方案）（很重要，必须做!!!）



6、大作业的提交要求

Step-2：删除BigHW目录下的 .vs 和 Debug 两个目录（很重要，必须做!!!）

★ .vs 和 Debug 存放在编辑/编译/运行过程中产生的各种临时文件

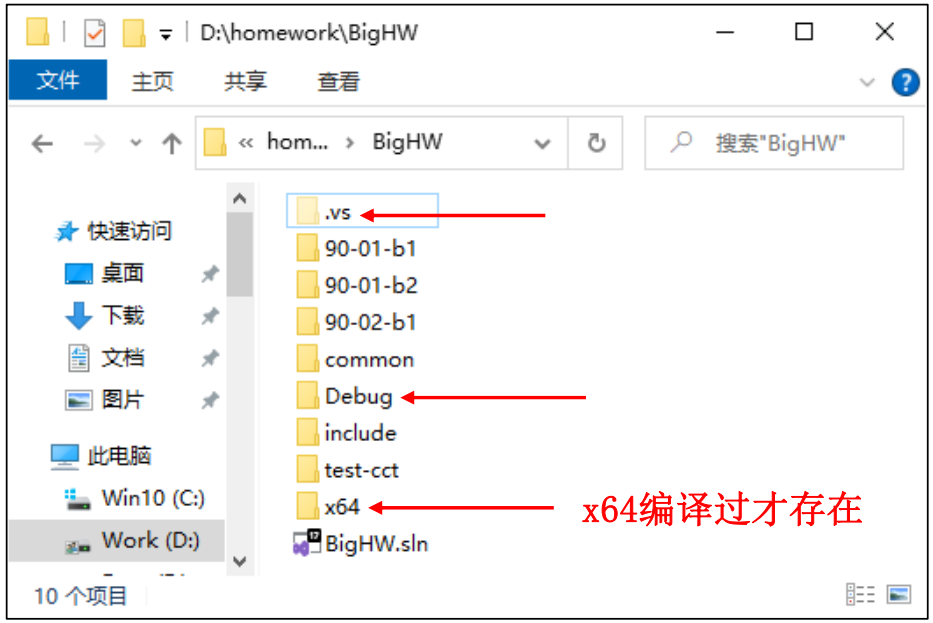
★ .vs是隐含目录，只有在控制面板的资源管理器属性中设置显示隐藏文件后才能看到

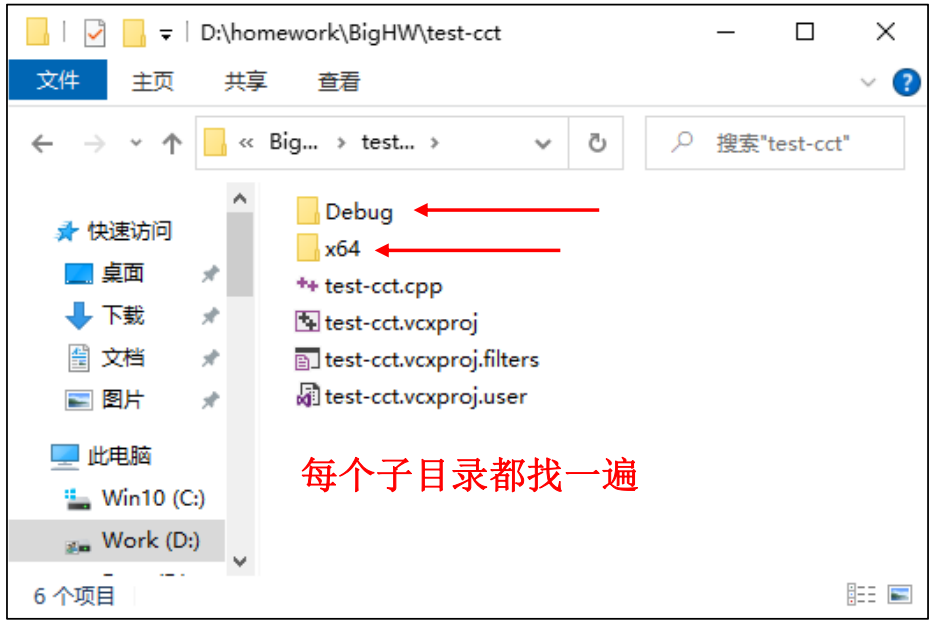
★ Debug目录中存放编译过程中的各种临时文件及可执行文件

★ 因为 .vs 和 Debug 两个目录很大(.vs可能到GB级别)，所以提交时要删除这两个目录

★ 如果BigHW及各子目录下还有x64，说明曾经用x64方式编译过，也要删除(未编译过则不存在)

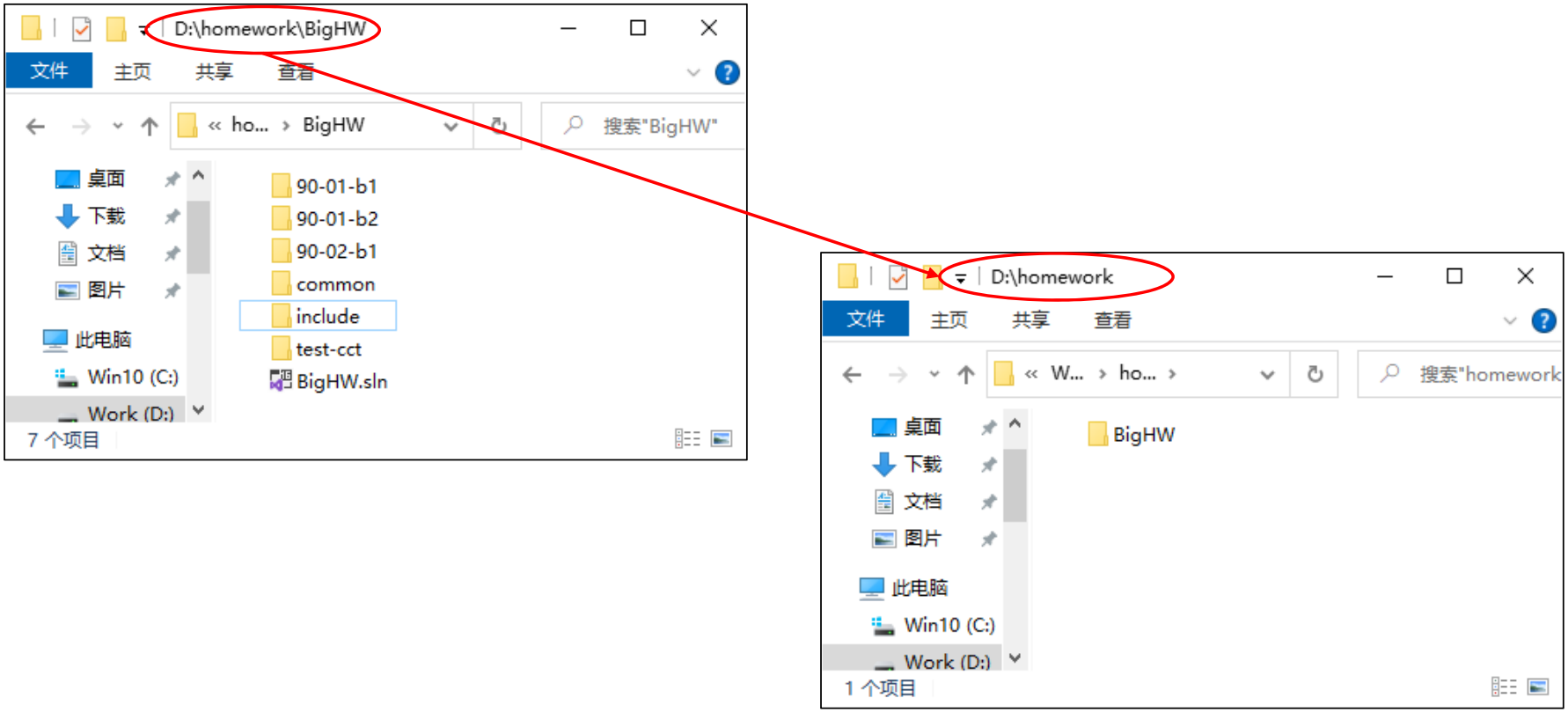
★ 各子目录下面的Debug/x64也要删除





6、大作业的提交要求

Step-3：删除 .vs 和 Debug 后，退回上层目录



6、大作业的提交要求

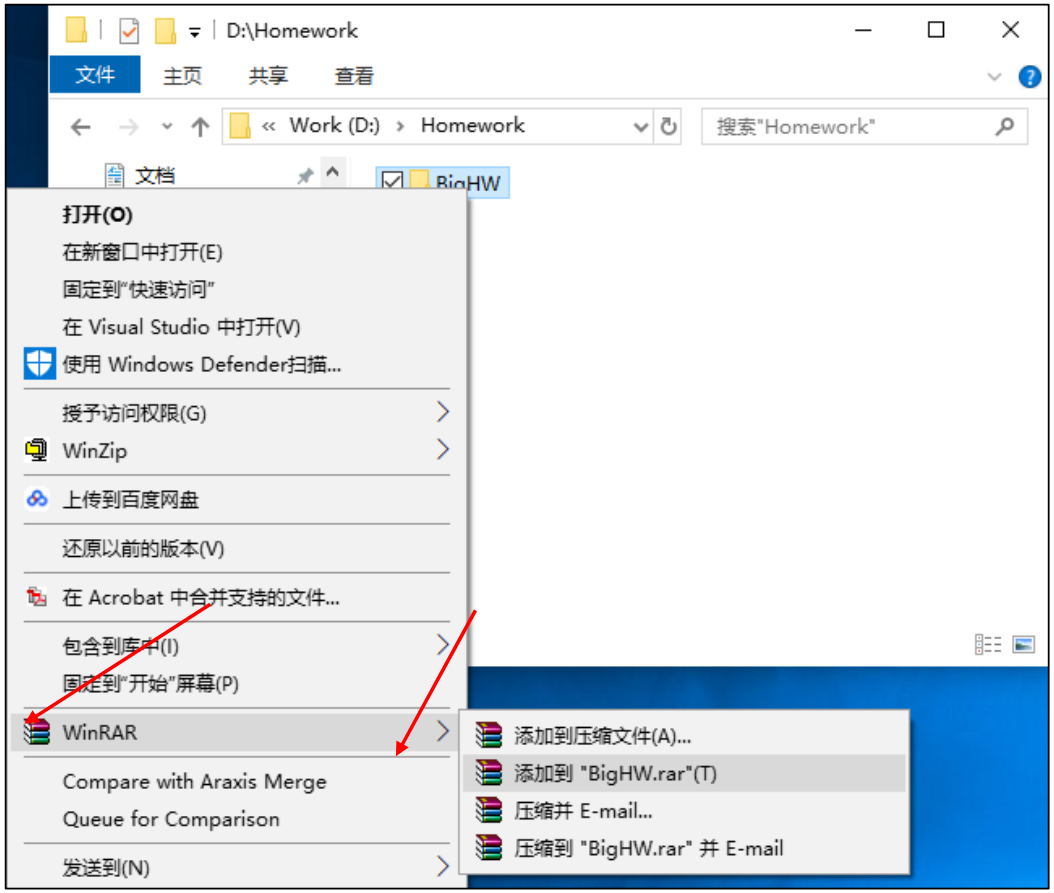
Step-4：将 BigHW 目录压缩为 BigHW.rar 并改名提交

★ 图示为使用winrar进行压缩，如果用其他压缩软件，要保证格式是rar(检查时用winrar解压)

★ 将生成的 BigHW.rar 文件改名为本次作业要求的提交名称(例：90-02-b1.rar)后提交即可

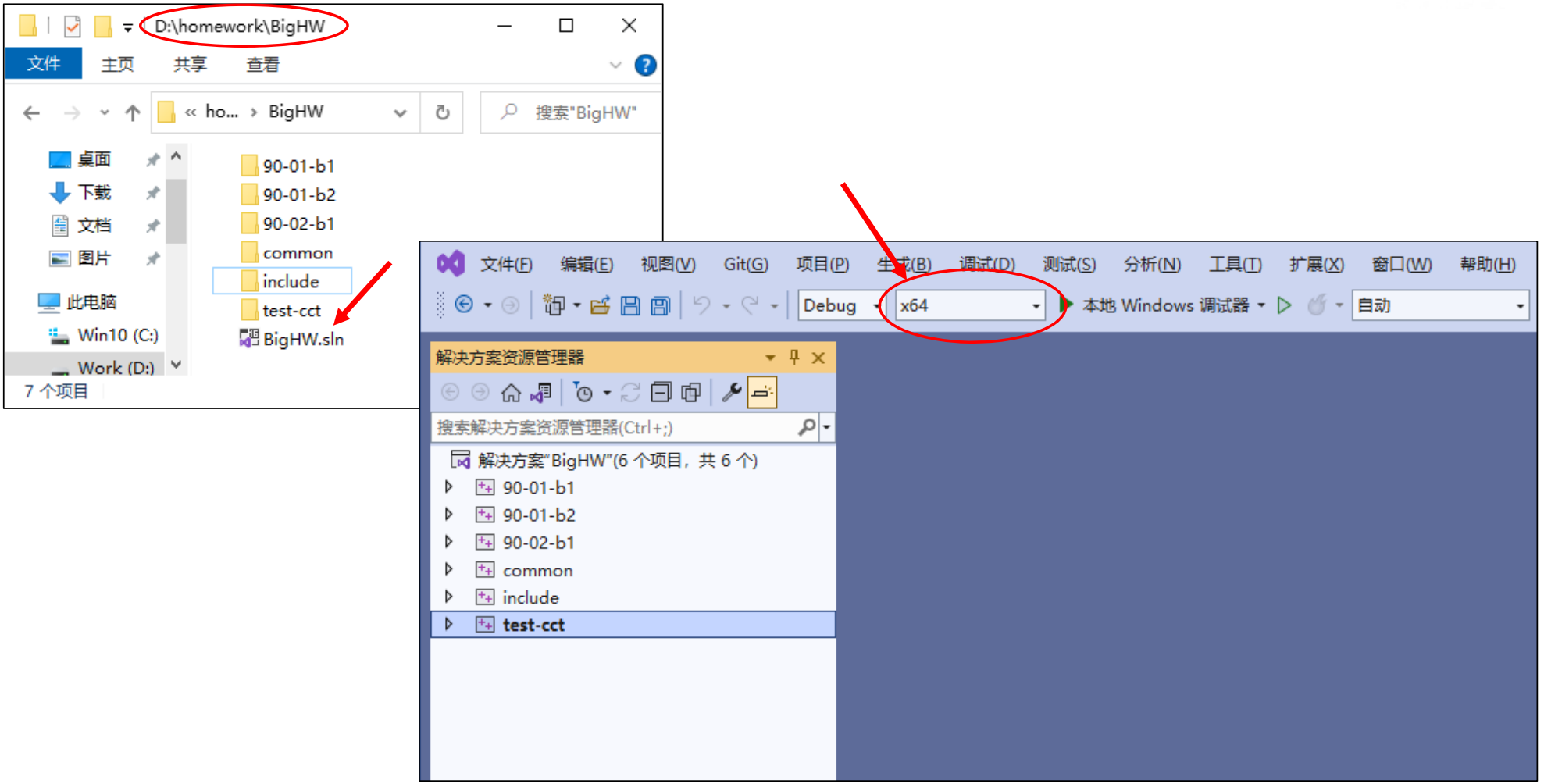
★ 要保证提交文件解压后有且仅有一个BigHW目录，且BigHW目录中是各项目的子目录(否则零分)

★ 在删除 .vs 和 Debug 目录后，压缩包的大小应不超过1MB



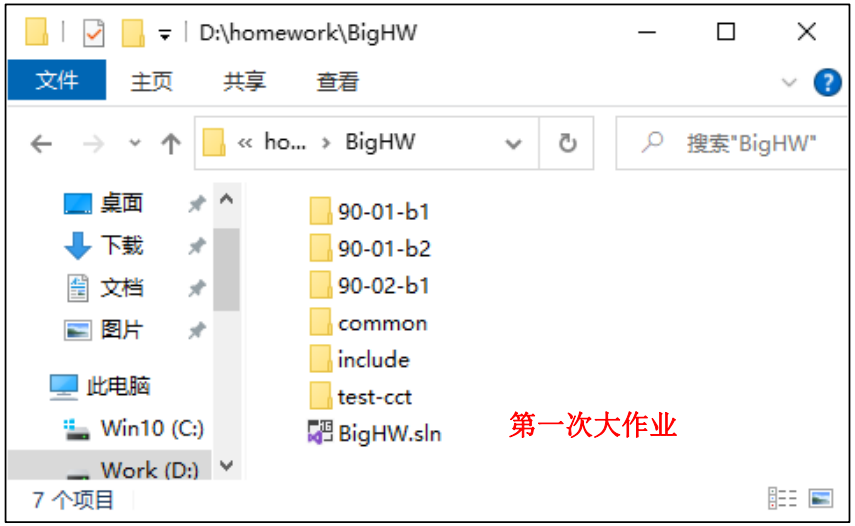
6、大作业的提交要求

Step-5：(特别提醒!!!)BigHW目录中的.vs和Debug被删除后，再打开解决方案，会恢复为x64，要再次设置为x86

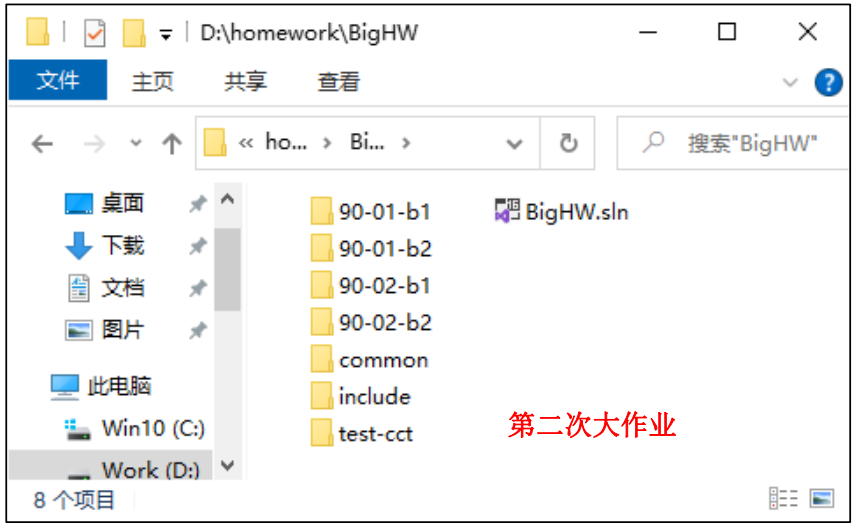


6、大作业的提交要求

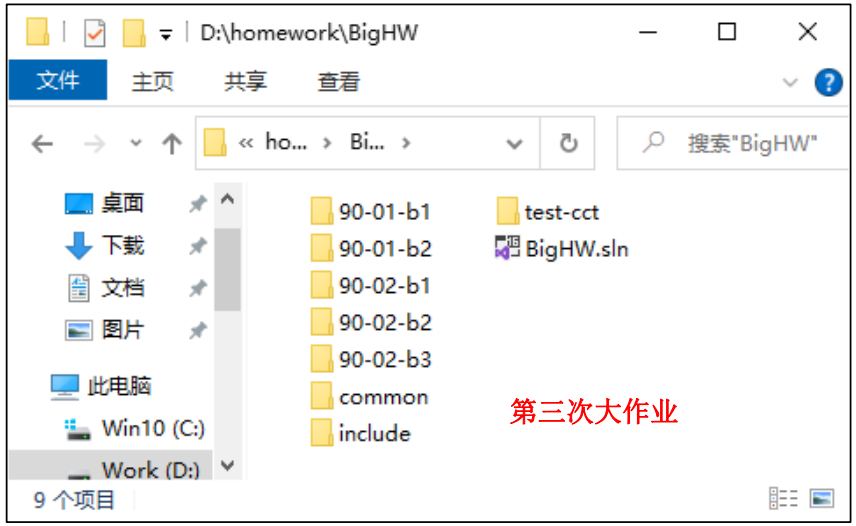
Step-6：(特别提醒!!!)大作业每次提交，均为BigHW目录压缩后提交，只是其中的项目数量会慢慢增加



第一次大作业



第二次大作业



第三次大作业

7、大作业的评分标准及注意事项

★ 不允许将公共文件复制到各目录下由不同项目分别使用，即cmd_console_tools.h/cmd_console_tools.cpp等文件
不允许在test-cct/90-01-b1/90-01-b2/90-02-b1等目录下出现

★ 每个大作业提交时，必须保证之前的大作业均能编译通过并运行正确，否则之前的得分会被扣除
（工程理念：公共代码的后续改动导致前面已通过测试的代码出问题是不可接受的!!!）
（例：提交90-02-b2时，要保证90-01-b1/90-01-b2/90-02-b1编译通过并运行正确，否则90-02-b1等之前的项目会被扣分）

★ 每个大作业必须在规定时间内完成，后续修改正确不加分
（例：90-02-b1正常提交为60分，90-02-b2提交时，顺便改进了90-02-b1，则90-02-b2按正常评分，90-02-b1即使改的比60分高，最终得分仍为60）

★ 允许随着作业进度而不断修改公共函数的名称/形参/具体实现等，只要保证正确性即可
（例：提交90-02-b2时，90-02-b1/90-02-b2合用了公共函数fun，在90-02-b3布置后，将fun改名为function，并修改了形参个数/类型/具体实现等，只要统一修改了90-02-b1/90-02-b2/90-02-b3中调用该函数的部分并保证编译及运行正确即可，鼓励此做法!!!）
（例：提交90-02-b2时，将90-02-b1原来单独使用的某函数改为公共函数并放于common目录中，只要保证90-01-b1/90-01-b2编译及运行正确即可，鼓励此做法!!!）

★ 在最后一次大作业提交完成后，所有大作业均进行编译运行并测试通过后，还要看公共函数提炼的合理性/函数及源文件命名的规范化等，如果完成的好，会得到额外积分，分值另定
- 此项得分为人为判定，提炼公共函数程度越高则得分越高，但会存在一定的得分误差
- 为了尽量减少人为判定的误差，建议在源程序中多写注释语句
- 不承诺一定会给予额外加分，不承诺额外加分的分值，但承诺公平、公正

8、后语

★ 所有的限制、扣分要求，最根本的目的不是为了拉低同学们的得分，而是工程习惯的培养

- 尽量提炼公共函数，用不同参数来区分细微的功能差别
- 形成自己的一套对独用/公共函数进行合理命名的规则
- 养成一个项目包含多个源程序文件、一个源程序文件包含若干同类函数的习惯，以及一套对文件进行合理命名的规则
- 取得一些在不断提炼公共函数时，如果保证使用公共函数的多个项目正确性的经历和经验
- 初学者肯定会在公共函数提炼，特别是修改后续作业后保证前次作业的正确性方面会遇到很多问题
  - 也许不提炼公共函数，复制粘贴后略作修改是最快的方法
  - 也许将某个独用函数提炼为公共函数后，会导致之前正确的程序出现错误并且无法及时准确地定位错误而影响得分
  - 坚持下去，必有收获!!!

★ 提示：如果违反大作业的要求，每次扣分至少8分起!!!