实验一、初步了解计算软件

一、实验目的

通过本次实验初步了解 NTL 和 Sage，计算一些具体的实例，为后续的实验奠定基础。

二、实验指导

NTL是一个数论运算C++库；Sage是一个支持代数、几何、数论、密码学、数值计算和相关领域的研究和教学的开源数学软件。以下的说明默认各位使用的电脑是Windows系统并在C++课上学会了使用VS编写C++代码。

NTL 官方网址：https://libntl.org/

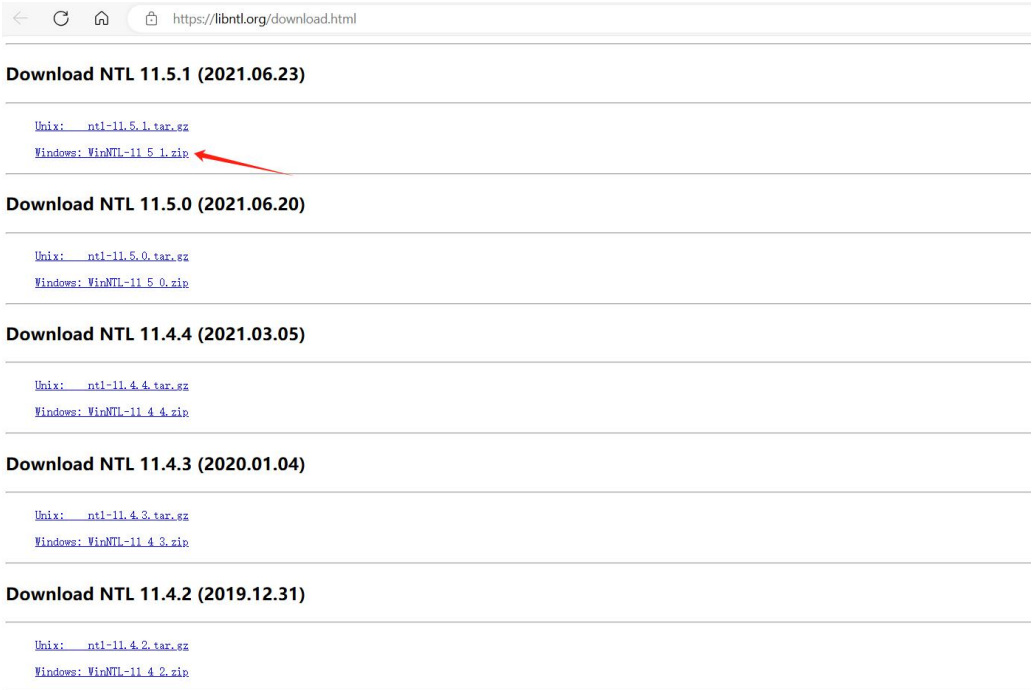
Sage 官方网址：https://www.sagemath.org/

（1）在电脑上的 VS 中调用 NTL 库：

在VS中调用C（++）库，相较于python调库会麻烦一些。以NTL为例，大体需要两个部分：配置NTL 库以生成NTL.lib 文件（一到五步）；在代码项目中附加包含上NTL代码并链接上NTL.lib库文件（六到九步）后使用NTL库写相关代码。如过程中遇到问题可到跳到第十步进行问题自查。

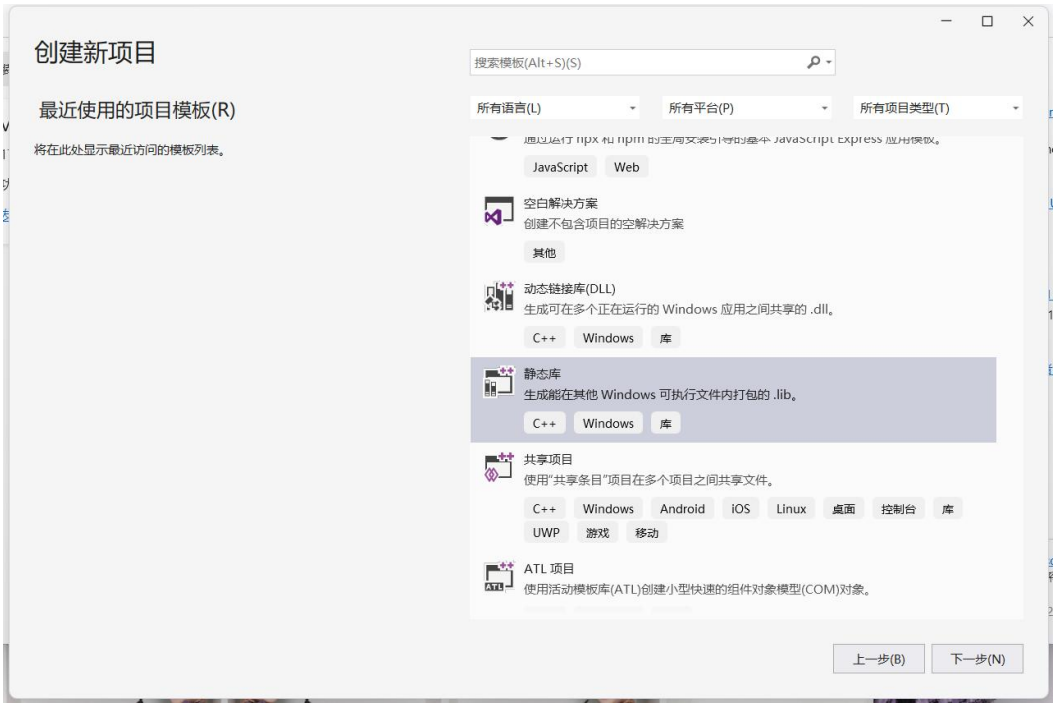
第一步：下载NTL库文件

在Download NTL (libntl.org)中或其他方法（镜像网站）下载winNTL库文件。



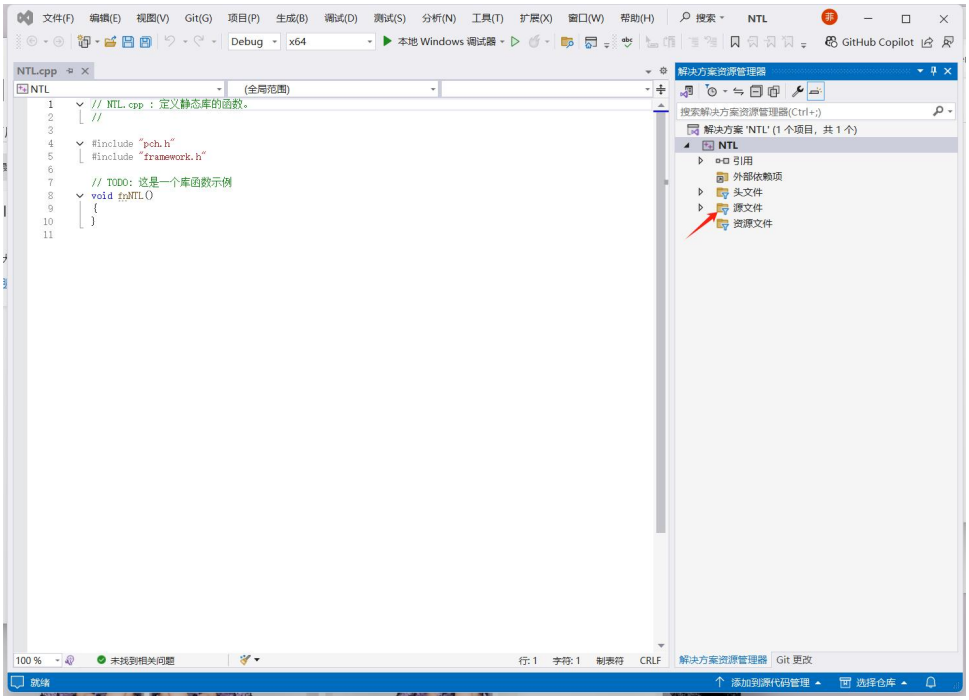
并将其解压到一处你可以记住的位置。

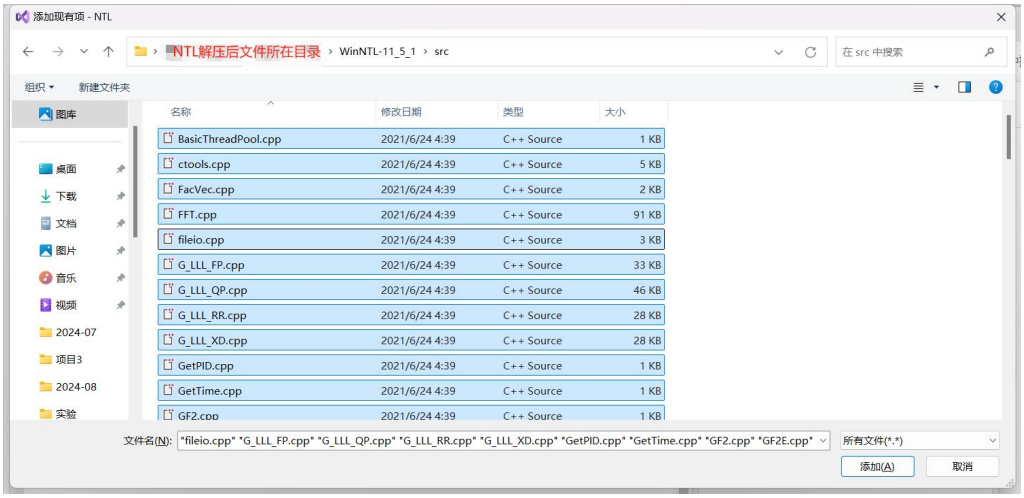
第二步：在VS中新建一个静态库项目，这里将其命名为NTL。



第三步：为项目添加源文件

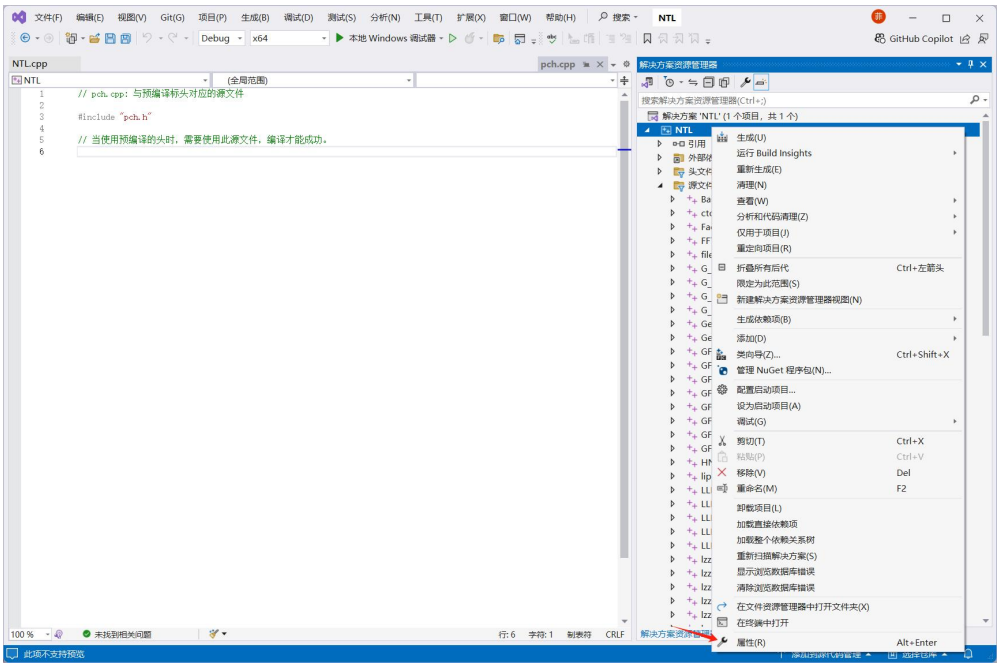
右键工程NTL->添加->现有项，选择WinNTL-11_5_1\src下全部源文件，(ctrl+A可快速选择全部源文件)，点击“添加”，即可将源文件加入工程。



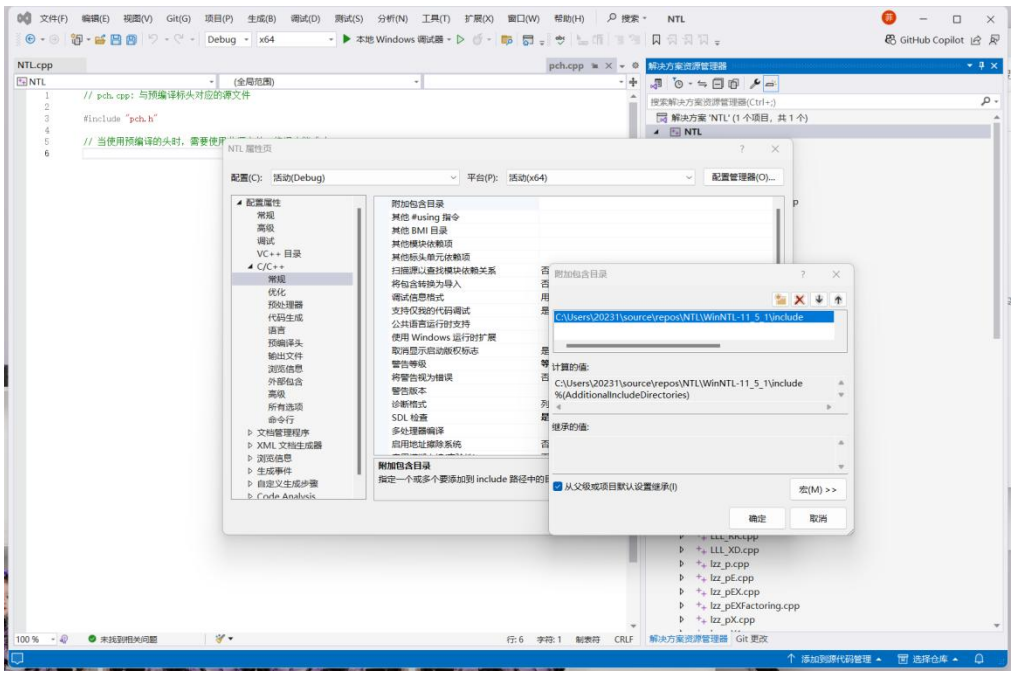


第四步：配置静态库项目属性

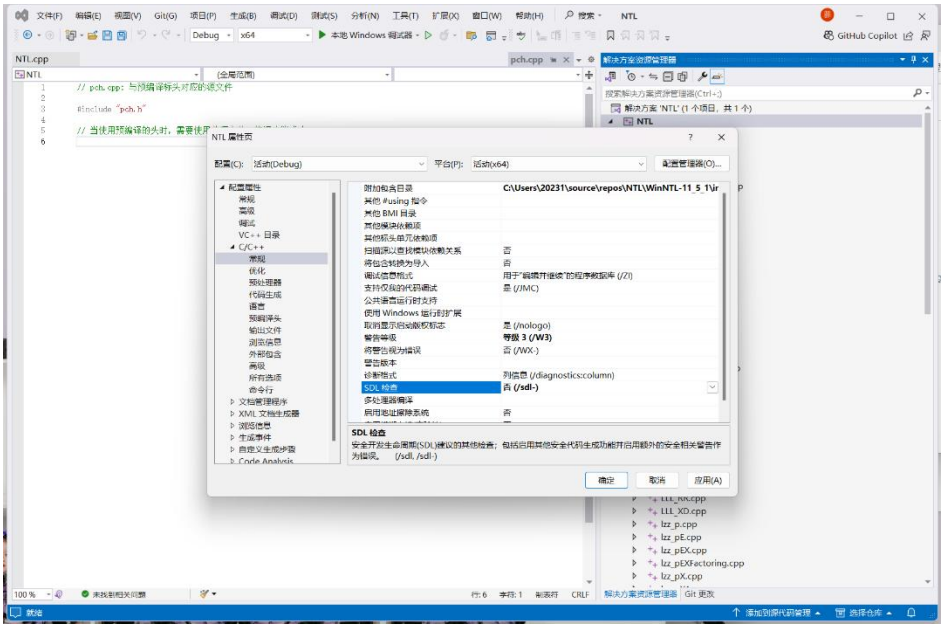
右键工程->属性->常规



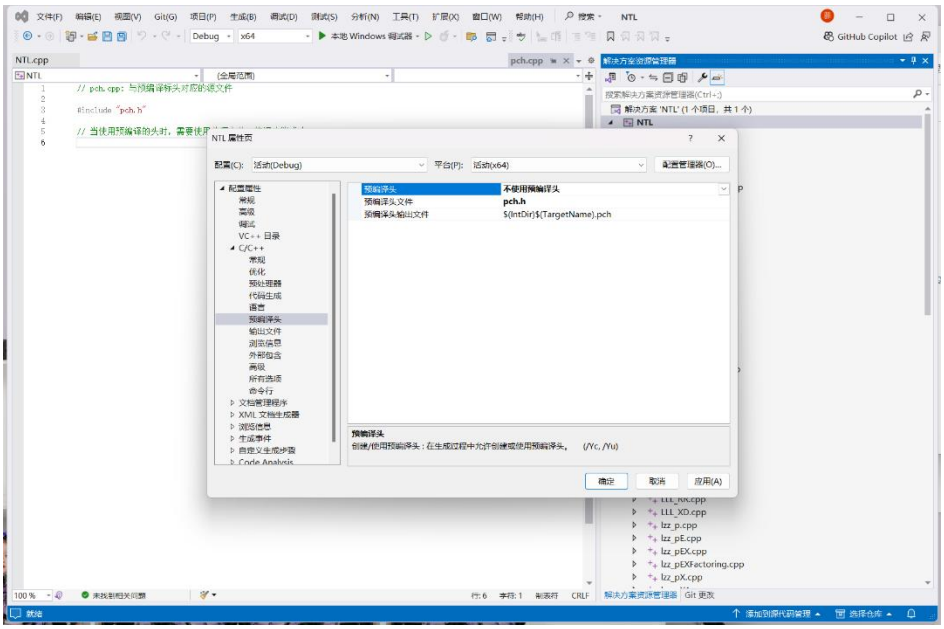
① 属性->C/C++->常规->附加包含目录，添加WinNTL下include文件夹的路径,选择从父级或项目默认设置继承



② SDL检查改为:否

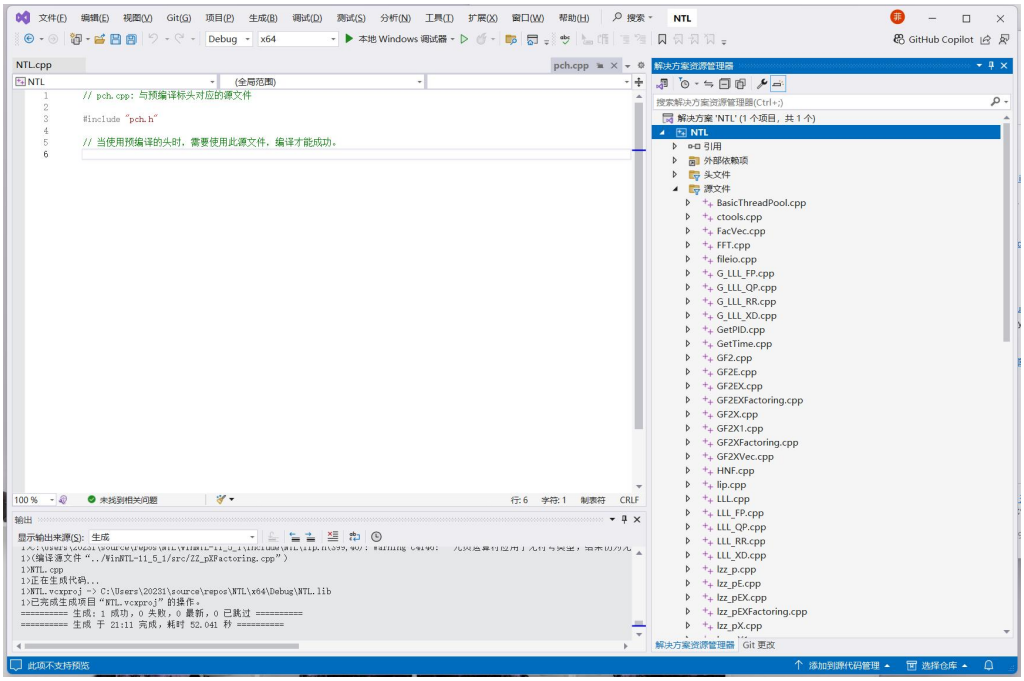


③ 预编译头改为:不使用编译头

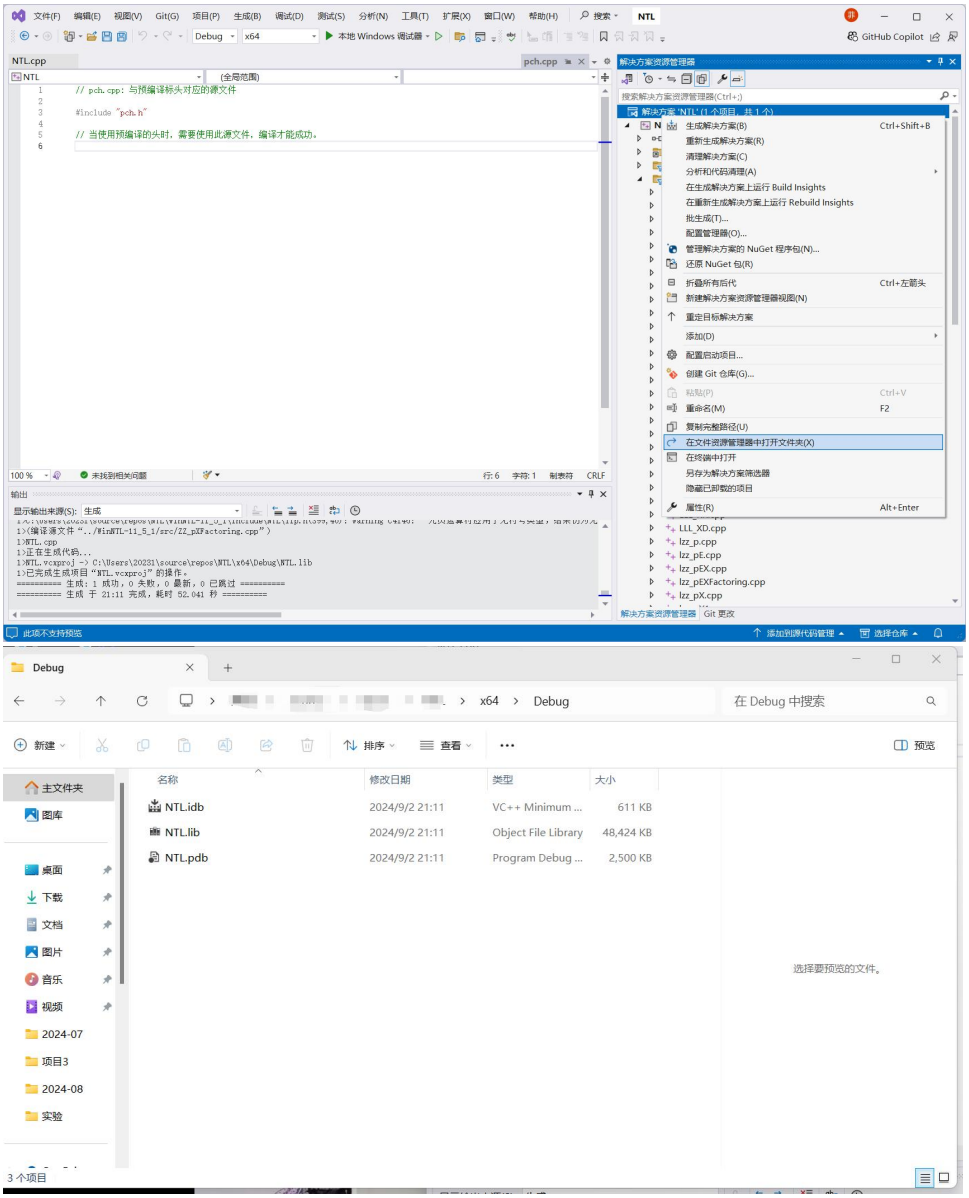


第五步：编译项目

右键项目->生成，发现项目运行成功!



查看项目文件，解决方案目录（默认情况为项目目录的上级目录）->x64->Debug(如使用x86和win32则在解决方案目录->Debug)下，存在NTL.Iib文件，说明项目WINNTL静态链接库创建成功!



第六步：在VS中新建一个空项目，这里将其命名为test。

第七步：新建源cpp文件，编写测试代码

可以使用如下测试代码：

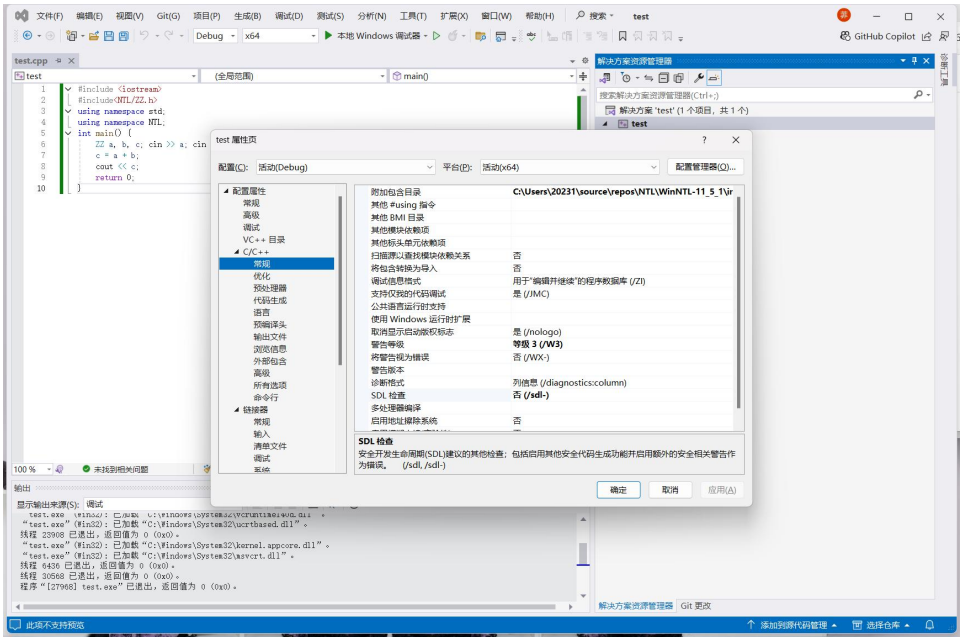
```
#include <iostream>
#include<NTL/ZZ.h>
using namespace std;
using namespace NTL;
int main() {
    ZZ a, b, c;
    cin >> a;
    cin >> b;
    c = a + b;
    cout << c;
    return 0;
}
```

第八步：配置项目属性

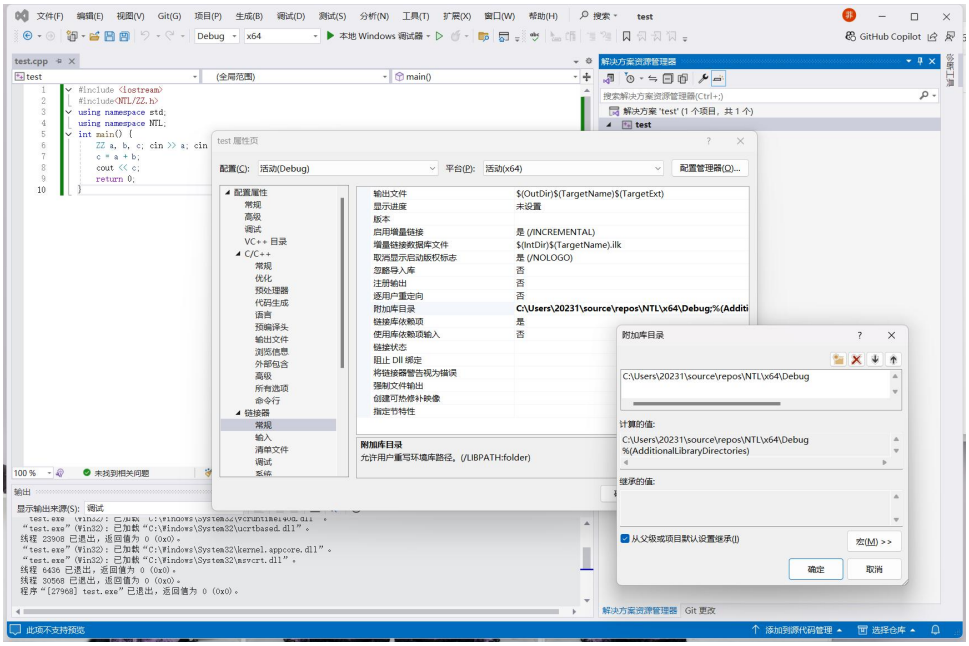
右键工程->属性

① 附加包含目录: WinNTL下include文件夹的路径

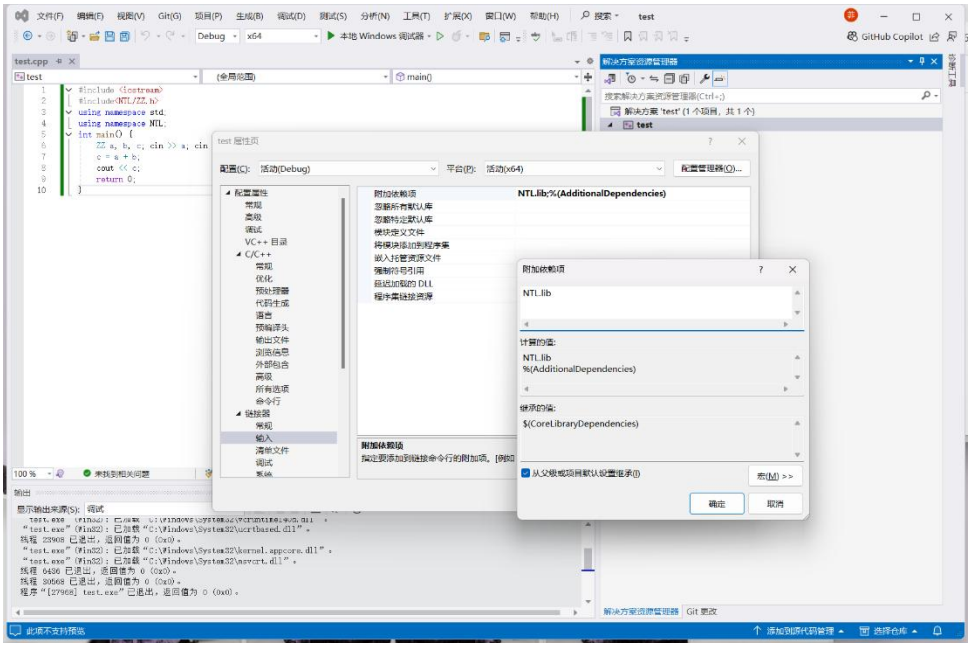
② SDL检查改为:否



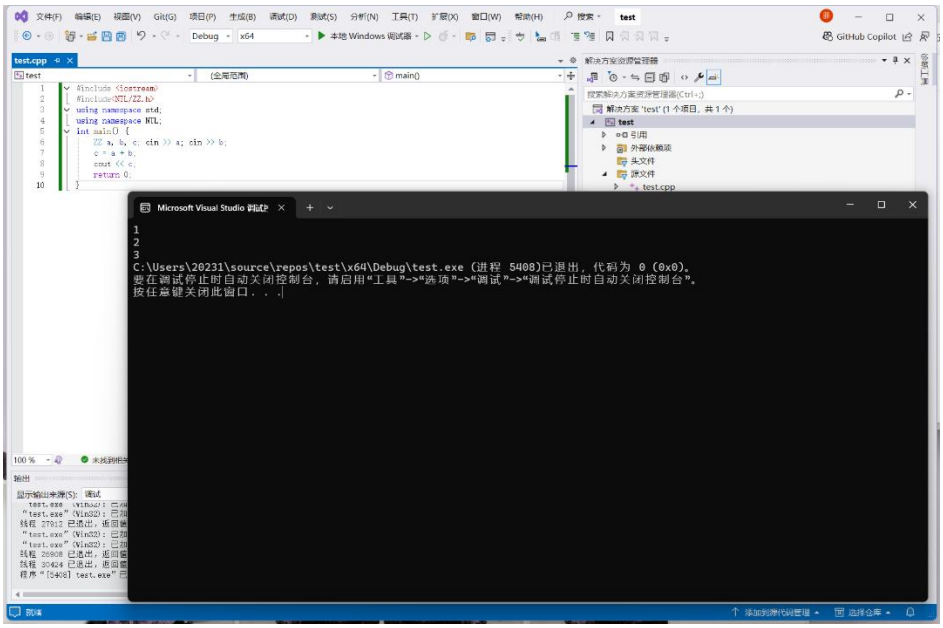
③ 属性->选择链接器->常规->附加库目录，选择 NTL.lib 文件所在的路径



④ 选择输入->附加依赖项，加上NTL.lib，选择确定



第九步：运行测试代码

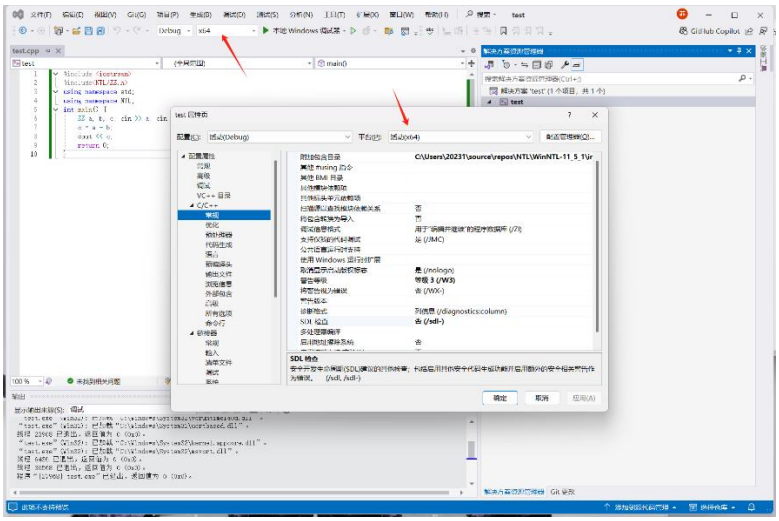


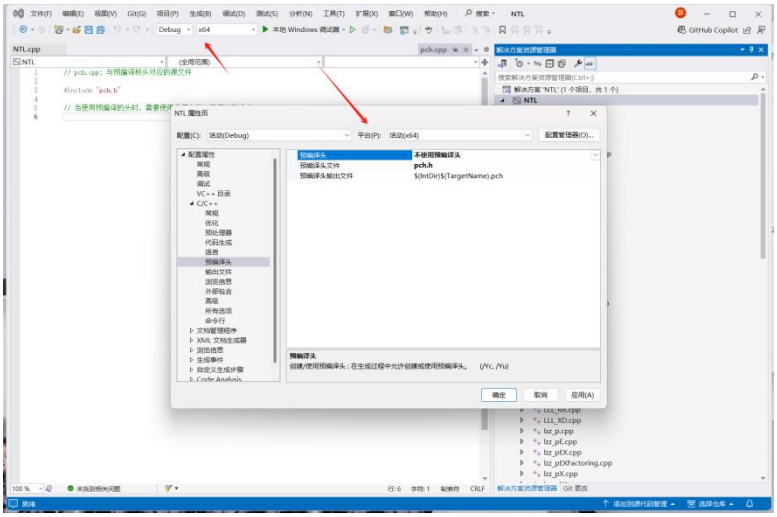
正常运行，调用NTL库完成。

第十步：问题自查

如前九步没有遇到问题的同学可以跳过这一步。

在整个调库过程中，需要保证如下两对位置始终保持一致，如截图演示部分一直保持debug与x64。并且编译库与使用库的两个项目设置也得一致。同理，debug可都改为release；x64可都改为x86和win32（x86和win32这里等价）。首先一定要确认这四处匹配。





几种报错与解决方向：

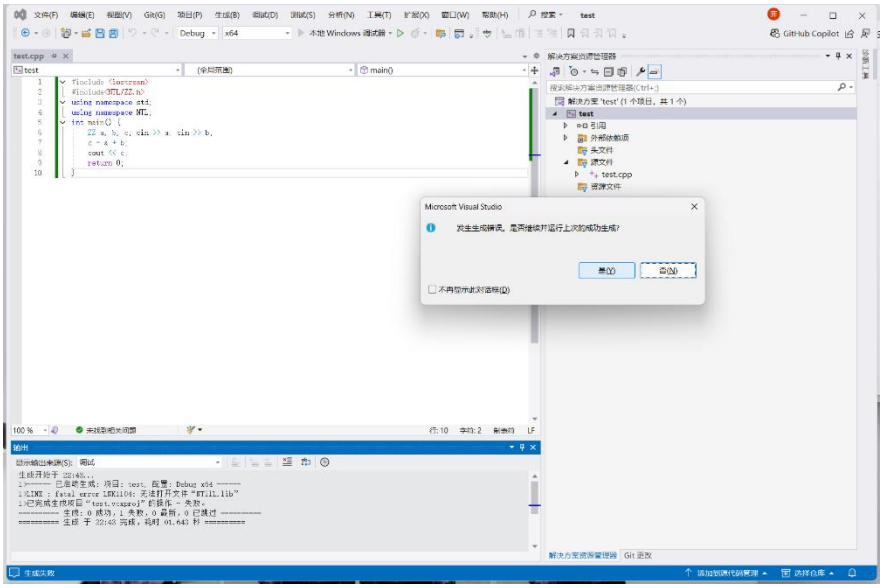
1、 编译库文件时弹出如下报错：

这实际上并没有问题，只是错误地尝试直接运行lib文件，检查NTL.lib文件是否已经正常生成，如已生成可直接正常进行下一步。

2、 代码NTL部分标错显示无法找到

检查附加包含目录include部分是否出现设置问题。

3、 代码没有标错但生成错误



若报错内容为

检查SDL检查改为:否

若报错内容为无法找到NTL.lib

检查链接NTL库部分是否出现设置问题；检查NTL.lib本身是否存在问题；

4、 可以生成但无法运行

检查NTL.lib本身是否存在问题；检查编译NTL.lib的过程。

(2) 在电脑上安装Sage：

Sage在9.3版本以后就不支持Windows上的直接安装了。为了图方便，可以直接下载并安装老版本的exe文件。

可以从镜像网站SageMath Download - win (aliyun.com)下载并安装Sage 9.3版本。此方法参考博客【SageMath】SageMath在Windows系统下的安装_sagemath下载-CSDN博客。

若是在Windows上想安装高版本的Sage，那就需要配置Linux虚拟机或Sage官方推荐的WSL（Windows Subsystem for Linux），在Linux虚拟机或子系统上就可以如一般的Linux系统的电脑一样安装最新版的Sage了，理论上执行

apt install sagemath

即可完成安装。至于完整的过程，建议同学们自行上网搜索。

三、实验内容

（1）安装好之后，结合文档进行练习：

NTL 文档： https://libntl.org/doc/tour.html

例如通过文件 Programming Interface 和 Summary of NTL's Main Modules 初步了解 NTL 的数据类型和接口， 通过 Examples 里的例子练习一下。

Sage 文档： https://doc.sagemath.org/pdf/en/a_tour_of_sage/a_tour_of_sage.pdf Sage

更多文档： https://doc.sagemath.org/

提示：在文档中搜索关键词可以使用 Ctrl + F，然后在搜索框里输入关键词。例如计时函数， 搜索 time。

（2）分别使用 NTL 和 Sage 具体计算如下 RSA 的参数生成:

(2.1) 选取两个随机的 1024 比特的素数 p,q;

(2.2) 计算二者乘积 N=pq， 测量所用时间；

(2.3) 选取参数 e=65537，测试是否满足(e,(p-1)(q-1))=1，不满足重新选取 e，如满足则 计算 d 满足 ed+x(p-1)(q-1)=1， 测量所用时间。

四、实验报告

完成实验报告，推荐学习使用 LaTex。实验报告主要包含如下部分：

(1) 报告题目，作者信息，每位组员完成的部分；

(2) 报告摘要：简述报告结果，例如可以做一个表格展示实验结果和时间；

(3) 正文：包括具体实验内容，相关的理论，结果的分析（例如时间对比）；

(4) 参考文献。

最后将实验报告和代码打包提交。 10 月 4 日之前将实验报告（命名格式：组号+实验 1） 电子版发给助教，邮箱：202337022@mail.sdu.edu.cn。

切记邮件主题和实验报告请按格式命名!