在线OJ注意事项

1. OJ概念

online judge，简称OJ，是一个在线的判题系统。用户可以在线提交多种程序代码(比如：C、C++、Java、Python等)，**系统对源代码进行编译和执行，并通过预先设计的测试用例来检验程序源代码的正确性。**现广泛应用于世界各地高校学生程序设计的训练、作业的自动提交判断，以及各种竞赛(比如ACM)等。

2. OJ原理

用户提交的程序在OJ系统下执行时将受到比较**严格的限制**，包括**运行时间限制**，**内存使用限制**和**安全限制**等。用户**程序执行的结果将被OJ系统捕捉并保存**，然后**再转交给一个裁判程序。**该裁判程序或者**比较用户程序的输出数据和标准输出样例的差别**，或者**检验用户程序的输出数据是否满足一定的逻辑条件。**最后**系统返回**给用户一个**状态**：**通过**、**答案错误**、**超时**、**超过输出限制**、**超内存**、**运行时错误**、**格式错误**、或是**无法编译**，并返回程序使用的内存、运行时间等信息。

3. 为什么要了解

由于在线OJ**成本低，出题判题方便，不受地域限制**，**节省时间与经济成本**等，目前**大部分公司招聘人才时选拔方式由线下开始转向在线OJ，**这就需要应聘者(尤其是即**将毕业的应届大学生**)对在线OJ的规则有一定的了解，才能尽情释放自己的所学。

4. 在线笔试流程

下面详细介绍在线笔试的**完整流程以及注意事项**：

- **投递简历**

  注意：**手机号、邮箱、求职意向等信息一定检查仔细，因为后续通知全是过邮件短提醒。**

- 笔试通知邮件和短信

  注意：如果收到短信没有邮件，可能是你的邮箱填错或者邮箱设置了拒收等原因，可以通过关注公众号：校招小管家 > 绑定收到短信的手机号 > 查询我的笔试。

- 检查考试设备

  1. 请使用**谷歌Chrome、火狐浏览器**访问笔试网站

     如果遇到页面加载不出来、摄像头不好使等情况，优先采取措施：**换另一个浏览器试一下。**浏览器下载地址：https://www.nowcoder.com/discuss/3793

  2. 确保电脑带有摄像头，并确保摄像头能够正常使用。

     摄像头检测：https://www.nowcoder.com/cts/3942933/summary#0

     - **摄像头黑屏、无法拍照等情况**

       优先换另一个浏览器，其次检查浏览器有没有adblock adguard等广告屏蔽插件，关闭后重试

     - **更换为前置摄像头**

       请点击地址栏右侧的设置--->高级--->隐私设置和安全性--->内容设置--->前置摄像头，进行调试即可

  3. 考试前**请关闭其他浏览器窗口，关闭QQ、微信、Skype等即时通信软件，关闭屏保，关闭Outlook等有弹窗提示消息的软件**，否则会被记录离开网页。

4. 确保网络连接畅通，网速应在100KB/S以上，**建议使用手机4G热点链接网页。**
5. 考试时允许使用草稿纸，请**提前准备纸笔。**考试过程中允许上测试等短暂离开，但请控制离开时间

- 笔试做题流程
  1. 试卷中会有一种以上个题型，进入考试后**请仔细查看共有几个题型。**
  2. 可选择任意题型进入做题，**所有题型一旦提交后将无法返回修改。**
  3. **可通过试卷页面底部答案卡进行同一题型试题切换**，但一旦进入某一类题型，提交后方可进入下一题型。
  4. 如**遇突发情况**，如断网、电脑卡死、断点等，请直接刷新页面，或**关闭浏览器后重新通过考试地址进入。**题目会自动保存，所以不同担心。
  5. 考试环境体验：https://www.nowcoder.com/cts/3942933/summary#

5. 在线编程题重点须知

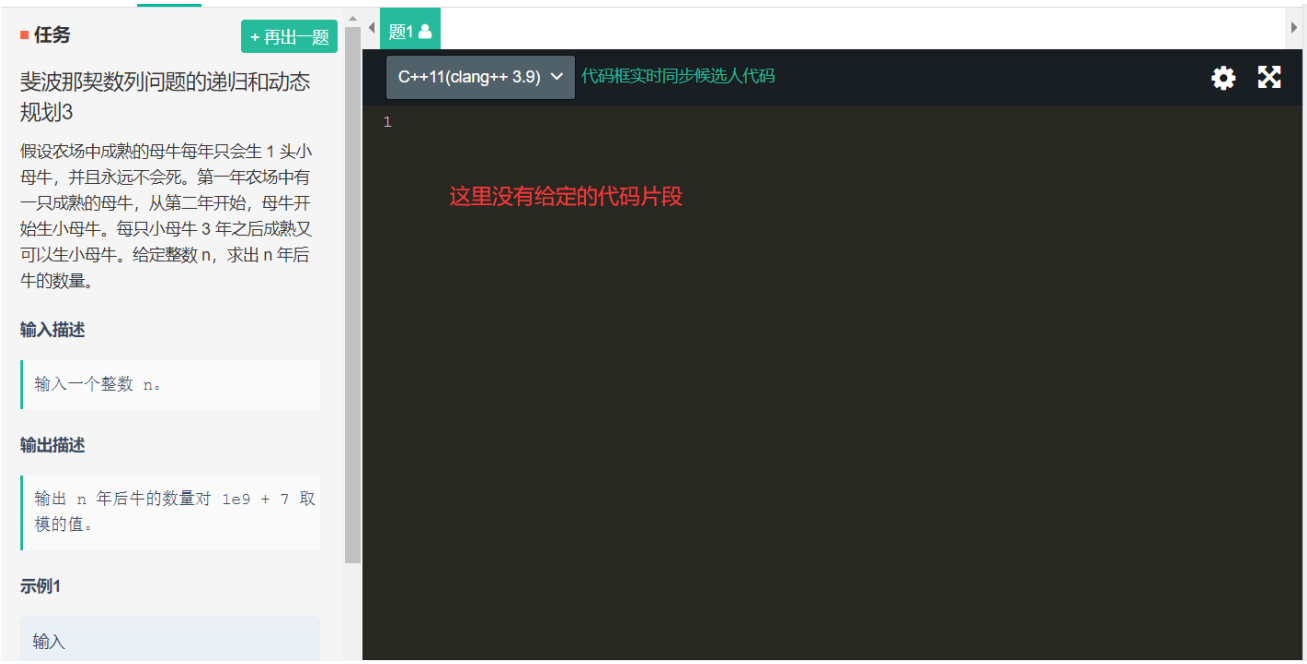
5.1 关于solution

认真做题的同学会发现，如果用C++写编程题的话，大部分平台会直接给你提供一个设计好的类(即一个class solution，里面有参数设计好的函数)，给了你函数接口，你只要实现函数功能就行。**为什么提交都是以class solution而不是main函数作为程序的入口呢?** 这是因为：

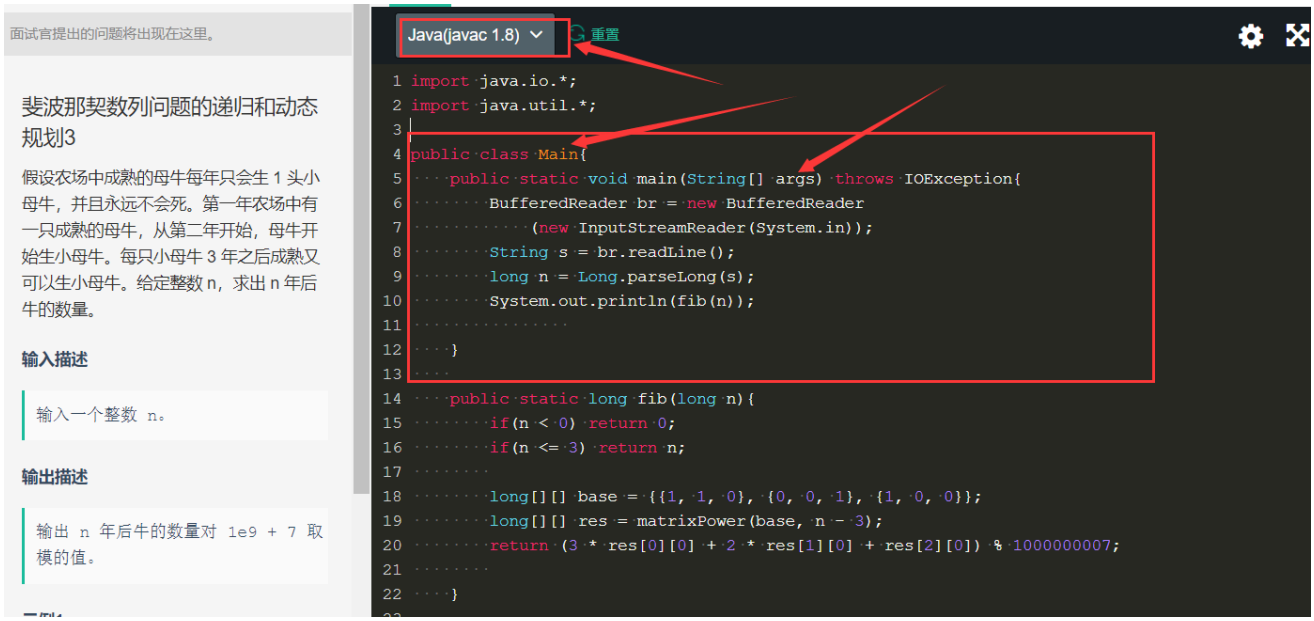
1. 其实**你提交的不是入口，main只是被隐藏**了。你提供的是一个封装了算法的函数。算法外的细枝末节不需要你操心。
2. 你的代码提交后，是要被单元测试的。要是写成main函数，就不方便测试了。
3. 因为一般OJ都会有时间限制，而 **cin 和 scanf 都很慢**，要求你写一个类的接口，**在评测的时候可以有效避免计算I/O的时间。**
4. **为什么要写一个类包含特定接口**而不是直接写一个函数，**是为了避免你写的其他函数和评测系统的函数冲突。**

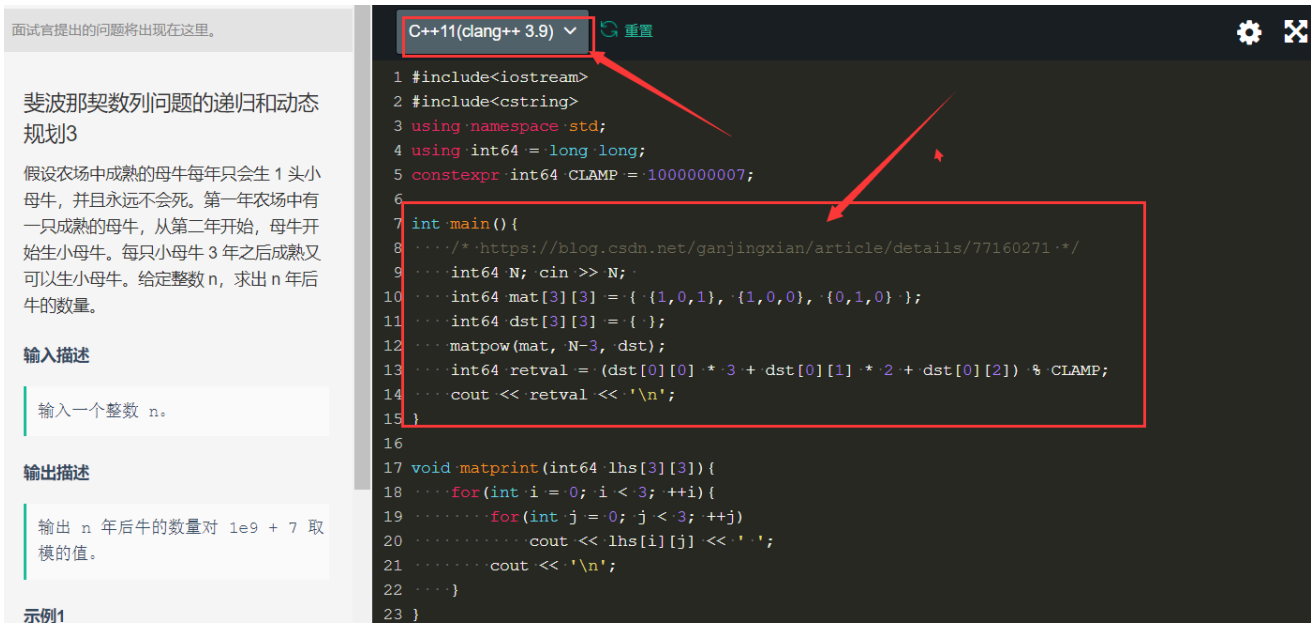
5.2关于答题时OJ窗口问题

有些题目打开之后，是空白的如：



答题方式如下：**Java方向类名必须是Main,且包含main函数，C++方向一定要包含main函数**





5.3关于输入输出

在线OJ对输入输出的格式有严格的判定，格式稍微不同(比如多一个空格)，就会造成代码提交不成功。

1. **正确处理输入格式**
    - **预先不输入数据的数组，处理方式：读到文件结尾**

```
//C语言
while(scanf("%d%d",&a,&b)!=EOF)
{
    printf("%d\n",a+b);
}

//C++
while (cin>>a>>b)
{
   cout<<a + b<< endl;
}
```

- **预先输入数据的数组，处理方式：读数组然后进行循环**

```
//C语言
scanf("%d",&n)
for(int i=0;i<n;i++)
{
    int a,b;
    scanf("%d%d",&a,&b);
    printf("%d\n",a+b);
}

//C++
cin<<n;
for (int i = 0; i < n; i++)
{
    int a, b;
    cin>>a>>b;
    cout<<a + b<<endl;
}
```

- **只有一组数据，处理方式：直接读取**

```
//C语言
scanf("%d%d",&a,&b);
printf("%d\n",a+b);

//C++
cin>>a>>b;
cout<<a+b<<endl;
```

注意：可以不用保存所有输入，读一组计算一组；字符串带空白的情况。

2. **正确处理输出格式**

   注意细节，看清题目要求。

   - **不需要输出case数**

```
cin<<n;
for (int i = 0; i < n; i++)
{
   int a, b;
   cin>>a>>b;
   cout<<a + b<<endl;
}
```

- **需要输出case数**

```
cin<<n;
for (int i = 0; i < n; i++)
{
   int a, b;
   cin>>a>>b;
   cout<<"Case"<<i+1<<a + b<<endl;
}
```

- **每个case之后有空行**

```
cin<<n;
for (int i = 0; i < n; i++)
{
    int a, b;
    cin>>a>>b;
    cout<<"Case"<<i + 1<<a + b<<endl<<endl;
}
```

- **两个case之间有空行**

```
cin<<n;
for (int i = 0; i < n; i++)
{
    int a, b;
    cin>>a>>b;
    if (i > 0)
    {
        cout<<endl;
    }

   cout<<"Case"<<i + 1<<a + b<<endl<<endl;
}
```

3. **行末空格**

   比如：输出需要打印多个数需要使用空格分隔的时候，我们循环使用 printf("%d ",x); 这种会很方便，但是这样会导致行末多一个空格，后台系统会严格比对你 的输出和.out 文件，这样也会被判错误。

4. **换行问题**

对于每个样例，建议输出完全之后都换行一下。对于一些题目，可能就是不换行就导致了后面输入数据错位，那就肯定不可能过了。

5.4 关于语言选择

做编程题强烈建议使用 C/C++，具体理由：

1. 出题人通常会使用 C/C++编写标程，数据也是由标程制造的，所以使用跟出题人一 样的语言会比较稳妥
2. C/C++效率比较高，通常来说一般 OJ 对于一道题目的时限限制会区分 C/C++和其他 语言，通常处理方式是假设 C/C++时限是 1s，其他语言就会给 2 倍时限，甚至更多。
3. 关于 cin cout 和 scanf printf。做题的时候尽量使用 scanf printf。下面告诉一 个小常识，不要惊讶：**cin cout 比 scanf printf 慢 20 倍左右**，一旦遇到大数据量，光是读入就有可能跪掉。
4. Java 相关：Java 整体效率大概比 C/C++慢 2~3 倍，但是 Java 写编程题也没什么问 题，主要就是处理好各种输入输出的情况。
5. python等等其他语言，做编程题真心不建议使用这些语言，要么效率低下，要么会有些更深的坑。

5.5 关于时间复杂度

通常来说一般的系统 1s 能跑的算法量级是不足 1e8 的，所以做题的时候评估算法效率很重要，直接判断你的做法能否通过，当然这是以 C/C++为标准的，其他语言自己乘个时间倍数。

举个例子，比如题目 n = 1e5，那么我就可以很敏感的知道我的算法需要一个 O(n) 或者 O(nlogn)。平方复杂度直接拜拜！

5.6 关于提交不成功

提交不成功，一般就两种情况：

1. 代码语法问题，导致编译不同过

   a.多去刷题，将常犯的错误积累下来

   b.避免使用一些奇怪的函数，或者与平台相关的函数

   c.最好不要使用vs来写算法，vs默认是Windows下的方式，一般OJ编译器可能无法识别

2. 代码编译成功，提交不成功

   a.查看输出不成功的信息，输出格式不对还是无法通过某个测试用例

   b.对于无法通过的测试用例，一般都会将无法通过的用例显示出来，然后提供一个本该输出与程序输出，通过输出结果以及测试用例，检查代码那块有误。

   c.实在检查不出来，逐个测试用例处理

3. 自己编译器可以执行，但是OJ不可以通过

   如有部分同学会说，在自己的编译器上可以跑通代码，但是在OJ上面不可以提交原因是什么？这种问题有很多种，比如**传入的参数**没有进行合法检查，例如：指针没有判空等等。这种问题大多是这些边界的问题。

4. 牛客网在线判题系统使用帮助

   <https://www.nowcoder.com/discuss/276>

5.7 关于作弊

任何情况下，都不要作弊，都不要作弊，都不要作弊。不要抱有侥幸心理，一旦被检测到，即使成绩再高，也会作废。在线系统如何防止作弊呢？

1. **限制切屏次数**

通过设置切屏次数防作弊，如果**考生在考试的过程中随意切换屏幕搜索答案**，如果超过一定次数，就被强制交卷。一般在强制交卷前都会有提示，如果还继续切换品屏幕，那将会执行交卷动作。

2. **防相互作弊**

   例如，优考试是按照如下方式处理的：

   - 时间限制，制定出相应的时间，只给出做题时间，不留任何作弊空间
   - 随机抽考，系统任意打算考试顺序，随机发放给每一个人
   - 试题不同，创建试卷时，选择随机组卷功能，出来的试卷每份都不一样
   - **检测代码的重复度**

3. **视频监控**

   考试时会根据试题比例来给考生拍照，防止考生中途换人。

4. **手机被监控**

   屏幕被监控，也不能用手机搜索，有可能会被摄像头监控到，且有些在线笔试可能会有二维码，考试之前会要求考生用手机扫描二维码，之后考生就不能用手机作弊。

5. **防死机、关机、掉线的人为或意外情况:**

   对于考到一半出现意外，考试系统会自动保存考试时间及答题内容，也就是说，假如设定考试时间为60分钟，学生只做到了一半就出现意外情况，花去了30分钟，那么系统会给该账户自动保存，当下次再次登录时，考试花去时间及答题内容不发生变化。

再次强调**请不要作弊**，考试前确保以下事宜：

1. **摄像头良好。**
2. 考试前**请关闭其他浏览器窗口，关闭QQ、微信、Skype等即时通信软件，关闭屏保，关闭Outlook等有弹窗提示消息的软件**，否则会被记录离开网页。
3. **确保网络连接畅通**，网速应在100KB/S以上，**建议使用手机4G热点链接网页。**
4. 考试时允许使用草稿纸，请**提前准备纸笔。**考试过程中允许上厕所等短暂离开，但请控制离开时间。
5. **使用草稿纸时尽量在摄像头控制范围之内。**