

云南省青少年编程挑战精英赛

2024.10.12

1

本次比赛过程，完全模拟**CSP-J/S**第二轮和**NOIP**的要求。

一，赛前一天

- 1，考场门上贴上门贴
- 2，机房门口贴上座位对应表（机房已编号就不用再贴）
- 3，打印签到表
- 4，查收试题压缩包（不能打开）

二，比赛当天--赛前

- 1，组织学生进考场。检查双证：准考证、身份证。
- 2，迟到15分钟不能进考场（可适当放宽至20分钟）
- 3，进入考场：
 - 不能带草稿纸！
 - 不能带一切电子产品（包括U盘、电子手表都不能带）
 - 可以带水、食品。
- 4，负责人8：00左右接收密码：压缩包密码、PDF打开密码。

三，比赛当天--赛中

- 1， 8： 20， 向选手发放试题、公示密码， 在黑板上写明可以换组。
- 2， 详细演示文件保存方式。
- 3， 检查是否断网。
- 4， 8： 30， 开始比赛。比赛过程中建议再次检查学生证件。
- 5， 11： 00可以交卷。两个组别时间是一样。
- 6， 交卷时， 请让学生到讲台上教师机处， 检查： （1） 文件夹名是否正确， （2） 打开第一个程序确认是否是自己的程序。

关于 CSP-J/S 数据提交格式的说明

在向CCF NOI办公室提交CSP-J/S数据时，请各省（市、区）遵照本文档组织相关数据和文件。

特别提醒：（1）选手试题目录下，不得包含除有效答案以外的任何文件和目录；

（2）`namelist.csv`与`answer`目录中的选手编号必须严格一致。

1 数据清单

各省（市、区）应提交的CSP-J/S数据包括：

- 选手名单文件-----`namelist.csv`
- 包含选手答案的子目录-----`answers`

上述数据应存放在同一目录下。目录名由表1中的省（市、区）编码、短间隔线“-”，以及“Junior”或“Senior”组成：

- 1) 普及组 目录名 = 编码 + “-” + “Junior”
- 2) 提高组 目录名 = 编码 + “-” + “Senior”

以北京提高组为例，其提交数据的目录结构为：

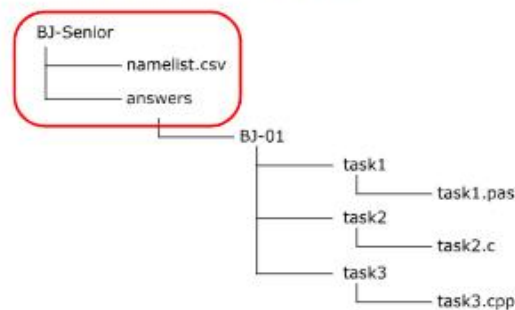
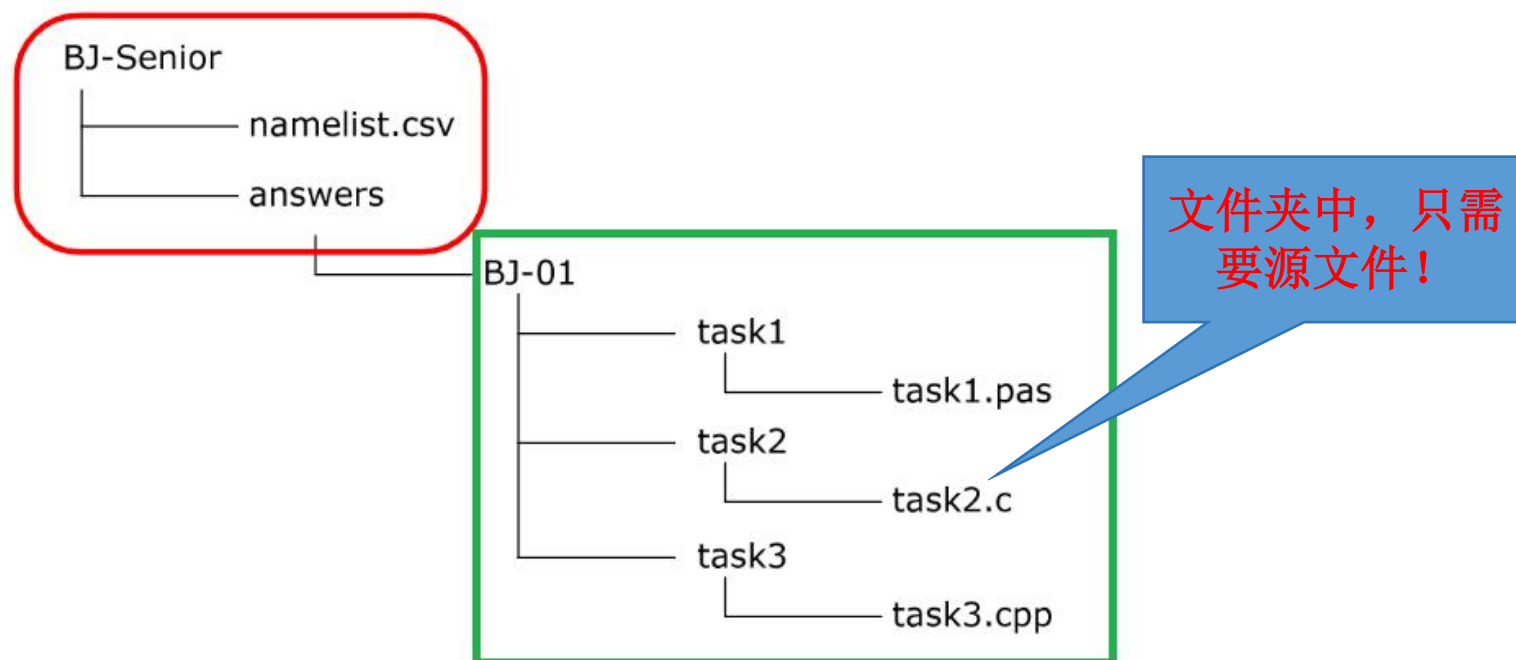


图1 提交数据的文件结构示例

隔线“-”，以及“Junior”或“Senior”组成：

- 1) 普及组目录名 = 编码 + “-” + “Junior”
- 2) 提高组目录名 = 编码 + “-” + “Senior”

以北京提高组为例，其提交数据的目录结构为：






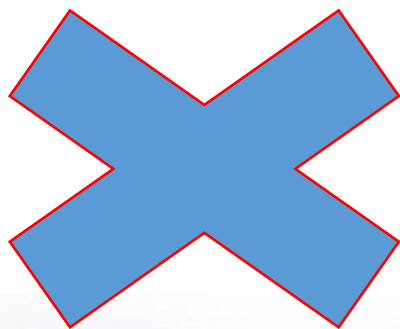
提交代码时注意：

- 1，文件夹用考号，不要写错！
- 2，考号文件夹里面，是四个题目的文件夹，每个题目的文件夹里面是一个对应题目的.cpp文件。
- 3，交卷时，提醒学生，除了.cpp源文件之外，其它文件不需要（最好删除）。

文件夹名字、层数一定要对！

昆二中 > JY-J0001

| <input type="checkbox"/> | 名称 | 修改日期 | 类型 |
|---|----------|------------------|--------|
|  | game.cpp | 2024/10/12 17:22 | CPP 文件 |
|  | lock.cpp | 2024/10/12 17:22 | CPP 文件 |
|  | tree.cpp | 2024/10/12 17:22 | CPP 文件 |



错误
!

换组

没有准考证号，用中文姓名建文件夹。

四，比赛当天--赛后

1， 14：30前，将所有代码压缩包发送到1741481283@qq.com
(线上考生，需在考试结束后10分钟内提交程序)

如果需要延迟时间，请在考点群里说明原因。

四，评奖及其它

- 1，成绩估计周二上午能出来。我们会尽快！
- 2，申诉及评奖等下周。

试卷

与CSP出题思路大体相仿

入门组：5题，前三题非常非常简单，后面两题有一定难度。估计会出现500分。

提高组：4题，CSP-S难度，没有简单题。

谢谢！

freopen()的七种错法

设定:

假设输入文件为lock.in,输出文件为lock.out, 题目给定的样例输入输出有三组: lock1.in、lock1.ans、lock2.in、lock2.ans、lock3.in、lock3.ans

假设输入：lock.in, 输出：lock.out

样例输入输出：lock1.in、lock1.ans、lock2.in、lock2.ans

正确做法（调试）：

调试时，读入数据写成

```
freopen("lock1.in", "r", stdin);
```

如果要输出到文件，写成

```
freopen("lock1.out", "w", stdout);
```

//lock1.out的文件名是随便起的，你记得打开它就可以看到输出。打开lock1.out与lock1.ans比对结果。

假设输入：lock.in, 输出：lock.out

样例输入输出：lock1.in、lock1.ans、lock2.in、lock2.ans

正确做法（提交）：

最后提交代码时，读入数据写成

```
freopen("lock.in", "r", stdin); //注意lock1.in中的数字1要删除
```

如果要输出到文件，写成

```
freopen("lock.out", "w", stdout); //注意lock1.out中的数字1要删除，  
后缀是out不是ans。
```

假设输入: `lock.in`, 输出: `lock.out`

样例输入输出: `lock1.in`、`lock1.ans`、`lock2.in`、`lock2.ans`

第一种错误: 函数名写错

`Freopen()`

`fropen()`

`freoepn()`

`fre()`

假设输入: lock.in, 输出: lock.out

样例输入输出: lock1.in、lock1.ans、lock2.in、lock2.ans

第二种错误: 数字忘记删除了

```
freopen("lock1.in", "r", stdin);  
freopen("lock2.out", "w", stdout);
```

假设输入: lock.in, 输出: lock.out

样例输入输出: lock1.in、lock1.ans、lock2.in、lock2.ans

第三种错误：输出的文件后缀忘记改了

```
freopen("lock.in", "r", stdin);  
freopen("lock.ans", "w", stdout);
```


假设输入: lock.in, 输出: lock.out

样例输入输出: lock1.in、lock1.ans、lock2.in、lock2.ans

第四种错误: 写完freopen()马上fclose()

```
freopen("lock.in", "r", stdin);  
fclose();  
//从文件读取数据
```

假设输入: lock.in, 输出: lock.out

样例输入输出: lock1.in、lock1.ans、lock2.in、lock2.ans

第五种错误: 双引号粘在一起, 参数间逗号没有

```
freopen("lock.in, r", stdin);
```

```
freopen("lock.in" "r", stdin);
```

假设输入: lock.in, 输出: lock.out

样例输入输出: lock1.in、lock1.ans、lock2.in、lock2.ans

第六种错误: stdin, stdout写错

```
freopen("lock.in", "r", std);  
freopen("lock.in" "r", instd);
```

假设输入: lock.in, 输出: lock.out

样例输入输出: lock1.in、lock1.ans、lock2.in、lock2.ans

第七种错误: 参数r和w用错

```
freopen("lock.in", "w", stdin);  
freopen("lock.out", "r", stdout);
```