**信奥机试注意事项**

**发试卷前**

至少提前15分钟到机房，确认设备是否正常。

1. 确认电源接触是否良好。
2. 确认键盘鼠标是否好操作。
3. 确认是否安装了DEV C++。
4. 做考前准备

**一、查看文件后缀名显示情况**

以“分糖果”为例，如果考试机隐藏了后缀名，有的同学不注意查看文件名容易保存为"airport.cpp.cpp"，这样文件名就不再是airport了，变成了airport.cpp为文件名了，与考试的要求不符。如何避免该问题？考前第一件事如下：

桌面上右击—“新建”—“文本文档“，如果看不见后缀名.txt，打开计算机，选“工具“—”文件夹选项”—“查看”—把“隐藏已知文件类型的扩展名”前面的勾去掉，就能见到“新建文本文件.txt”。

|  |  |
| --- | --- |
|  |  |
|  |  |

**二、配置编译环境**

1.选择C++14语法

“工具”->“编译选项”->“代码生成/优化”->“代码生成”->“语言标准”->”GNU C++11“。这样，DevC++可支持c++14的语法，例如auto关键字。

2.启用-Wall编译选项

不要忽略任何警告，没有警告信息能让你的代码更健壮。“工具”->”编译选项”->“代码生成/优化”->“代码警告”->“显示最多警告信息-Wall”->”Yes“。



3.使用万能头#include <bits/stdc++.h>要注意变量名与关键字冲突。

在DevC++环境下，用了万能头stdc++.h时，全局变量使用了c++14的某些关键字，如begin、end、y0、y1、yn、j0、j1、jn……若没有做上述设置Dev-C++不会提示，导致本地编译没问题，NOI评测时爆0。

|  |  |
| --- | --- |
| **没启用-Wall** | **启用-Wall** |
|  |  |

有一说法：“万能头文件**#include<bits/stdc++.h>**会带来一些诸如速度上的问题。”，这是指编译速度，他不影响执行速度。编译慢不会影响打分。Noi linux系统中，在终端里编译可以用如下命令：

g++ std.cpp -O2 std -g -fsanitize=address,undefined

运行时命令：

./std

这个会速度变慢，但检查一些下标越界等未定义错误

|  |  |
| --- | --- |
| 未用-g -fsanitize=address,undefined | 启用-g -fsanitize=address,undefined |
|  |  |

1. 编译优化选项。

启用“-O2”优化，工具”—”编译选项”—“编译器”—在“编译时加入以下命令”—“-O2”。启用“-O2 -g -fsanitize=address,undefined”亦如此。

1. **配置编辑环境**

工具—环境选项—基本—简体中文

工具—编辑器选项

**四、写好框架**

#include  <bits/stdc++.h>   //注意 / 不能写成 \，否则0分

using  namespace std;

int  main()

{

freopen("\*\*\*\*.in","r",stdin);

freopen("\*\*\*\*.out","w",stdout);

  //在本地调试的时候将freopen()函数注释，调试完毕后别忘记将注释删除

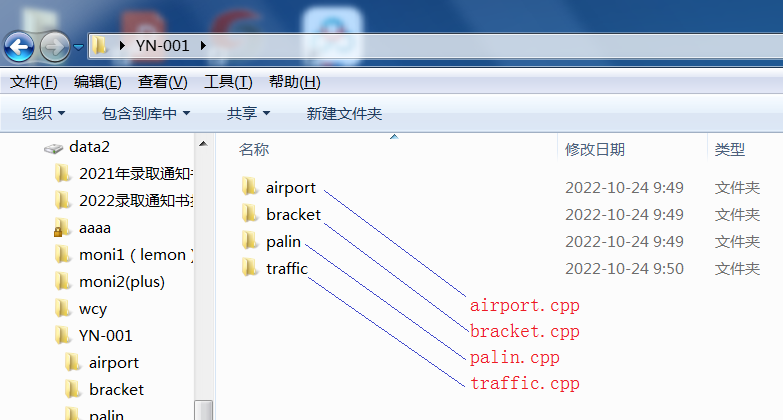
fclose(stdin);fclose(stdout);  //关闭文件流（可以不写）

return  0;

}

**做 题**

1. **试卷发了之后仔细阅读第一页**
2. 开始比赛，不要急于看题目，将试题第一页的时间、内存限制等等一字不落地看完，把每题的时空限制写在草稿纸上，建好文件夹结构。写题目名字的时候大小写是敏感的，一般是小写（千万不要写错，认真看，仔细写！）。 听从监考老师要求，根据老师的要求存放文件。（一般是在 D 盘根目录下，以 YN-XXX为名字建立一个文件夹，文件夹下分四道题目的文件夹，这四个文件夹内各自存放一个源程序。）



|  |  |
| --- | --- |
|  |  |

三、看题目时不能走神，看完题目后将其归入某几个框架中，包括:模拟/枚举/搜索/贪心/动态规划/图论/分治请将题目通读完以后，再开始深入思考你认为最容易的一道题。

1、根据输入数据的范围大致确定算法复杂度，即使这道题再容易，也不要着急写代码，请先明确自己每一步要干什么后，再开始写，轻敌会是你最大的错误。

2、即使这道题看起来再没法做，也不要提早放弃，这个时候纸和笔会是你最好的朋友，自己尝试几个例子，也许你就会找到答案。

3、请一定先明确自己要干什么之后再写程序不要走一步想一步写下代码前，必须保证有充足的思考时间，有成熟的想法后再动手。写代码前，尽量用多而强的数据去测试想到的算法，毕意代码写完后再测试就浪费很多时间了不能想一点写一点，就算是输入部分也要在整体思路理清后再写。

4、如果思考30分钟仍一头雾水，没有可以实现的算法，请你果断屏蔽掉100%的那一栏数据，开始写60%，50%乃至30%的算法--在机试里面，30分绝不是小数目。

5、如果你发现你旁边的人写得很快，请你放心，他的算法十有八九是错的。

6、虽然一般是1s+512MB内存，不过也不是每道题都是这样的，还是请认真阅读试卷首页的试题说明。

7、如果这是一道动态规划题，请先把转移方程写在纸上再编程。

8、计算内存的方法:数组大小\*类型长度/1000/1000=所占内存MB数，int类型长度是4字节，long long是8字节。

9、CCF规定源文件不能超过100KB，否则0分，所以使用打表法要特别小心。

10、计算数据大小，是否爆int，注意与int对应，与long long对应，改float为double ，与double对应。

11、涉及取模的问题，能多模不少模，涉及乘法建议强制转 long long 。（1ll\*n）

12、读写问题：

（1）读入到字符串，包含空格，C++14已弃用gets可以使getline(cin,s)，getline在cin之后可能会读到换行符。

（2）读写数据量过大，速度cin<scanf<getchar，那多大呢，读入1000000个数据cin需要700+ms，getchar需要100+ms

（3）输出数据量大时，以‘\n’代替endl，endl会刷新缓冲区，影响输出速度。

13、爆栈！！递归时可能会遇到，在Window下，栈的大小是2MB，Linux下，默认栈空间大小为8MB。

14、对定义的**变量**以及**数组**进行初始化。特别是在进行**累加、累乘**等操作的时候，一定需要对定义的变量或者数组进行初始化。栈中的变量（函数体中的自动变量）和堆中的变量（动态内存）会保有不确定的值；全局变量和静态变量static（包括局部静态变量）会初始化为零；当数组定义为局部变量时，如果太大将导致栈溢出。建议**数组长度超过100000时**，定义为**全局变量**，二维数组也是如此。

15、如果对大规模数据的算法正确性不敢保证。可分段处理:先写一个暴力程序应对小规模数据，再写一个高效程序应对大规模数据。以函数方式实现。这样也方便后续对拍。

16、写完程序之后，请一定不要忙着编译，请一定要将你的代码从头到尾通读一遍，也就是静态查错，这是整个编程过程中最重要的步骤，有的变量重复调用问题调试的话，一个小时也看不出来，静态查错可以一下指出错误。

四、静态查错请注意以下方面:

(1)是否写上了using namespace std?

(2)数组开得是否够大?

(3)变量类型是否正确，答案和中间结果是否会爆 int?

(4)memset时，所填的sizeof(X)的X是不是匹配?大小是不是正确?

(5)外层循环与内层循环的i、j是不是混用了?

(6)循环变量的枚举范围是否正确，是否会数组越界?

(7)涉及到边界处理、加一减一之类的问题请在纸上举个例子，标上下标以后，在编程时参照纸上的下标写。

(8)这个程序是在执行你想让它执行的步骤吗?

(9)读入优化处理负数情况了吗?在小规模数据时请不要速入优化反之则一定要速入优化。

五、通过样例后，请你一定不要放松警惕，因为样例并不能覆盖所有的情况，请自己设计几组大数据，争取卡死你的程序。特殊规律的数据，例如全是最小值，全是最大值，单调上升!下降数据最能影响程序效率与正确性。如果出现问题，请你调试你的程序，请一定要分模块调试，不要从头跟到尾。

* 如果你已经设计不出能卡住你的程序的数据，恭喜你可以做下一题了。
* 如果你用的是windows，如果用过，请你注意把system(“pause”)注释掉。
* 为了万无一失，请你用return 0;结束你的程序。
* 在内存允许的情况下，能开普通队列就不要用循环队列，能开普通数组就不要用滚动数组。
* 在时间允许的情况下，能暴力就暴力，高精度能不压位就不压位，优化不需要的就不要。

总之，在不超限制的前提下，能不优化就不优化，以减少代码量和出错概率为第一原则。

**提 交**

当比赛还剩下5~15分钟的时候，请不要再改动你的程序，即使你怀疑它对你的一个输入给出了错误答案，因为你自己算出的结果也有可能是错的。

这个时候请你检查是否注释掉了该注释掉（freopen）的东西，文件名是否写对，文件夹是否建对（每题只有一个文件夹一个后缀为.cpp的文件且只有一个），文件大小是否小于100Kb，请一定反复检查!

请记住，NOIP不怕暴力，怕瞎算，不怕不会，怕不敢，有时间的话一定要写一个暴力程序去对拍，验证算法的正确性。

沉着，冷静，没有思路上厕所。做能力范围内的事，尽力就好。不担心别人强大，小心自己失误。自己做得完美，就无怨无悔。

“厕所是 OIer 灵感的源泉。”洗手间可以让人冷静下来。

一定合理安排时间，不能炸心态，不能死磕一题。

补充：

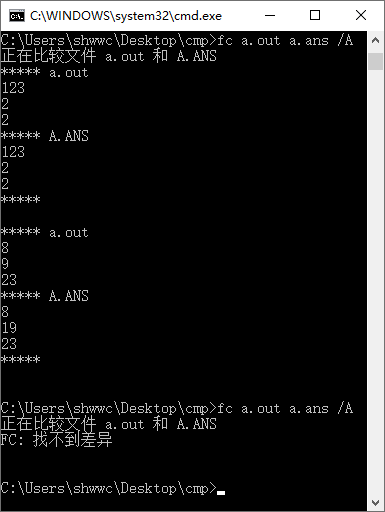
1、std影响

|  |  |
| --- | --- |
|  |  |

1. 13 E:\wcy\cpp\zmr\P1120.cpp [Error] reference to 'tm' is **ambiguous**

4 35 E:\wcy\cpp\zmr\P1120.cpp [Note] candidates are: int tm [70]

2、Windows中文件比较命令 fc



3、Linux文件比较命令

cmp diff