复旦大学计算机科学技术学院 专业硕士复试上机考试题（**2014-03-26**）

**1.** 按考场老师指示的方法和要求提交源代码文件。提交不必要的文件、建不必要的文件夹、不必要的压缩、不按要求命名文件，而导致评测程序找不到对应文件的，将得**0**分。没有特别说明的，输入和输出均为文本文件，存放在当前目录，即不要指定文件路径。文件命名如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 源代码文件 | | 输入文件 | | 输出文件 | |
| 第一题 | p1.cpp | | p1.in | | p2.out |
| 第二题 | p2.cpp | | p2.in | | p2.out |
| 第三题 | p3.cpp | | p3.in | | p3.out |
| 第四题 | p4.cpp | | p4.in | | p4.out |

2. 代码需使用标准**C**或**C++**语法，源代码内不要包含不必要的头文件(允许使用STL库)，一道题的所有代码都要放到同一个.cpp文件中，不要自己写其他头文件。程序的**main**函数请严格按照以下格式书写，不要使用编程环境自动生成的框架。程序入口必须为main，程序中没有main函数或不符合规定格式导致源码无法被机器正确评阅，将作0分处理。

//只包含必须的头文件

//不要使用IDE自动生成的代码框架

intmain() {

//此处填写代码

return 0;

}

3. 我们基本采用黑盒测试，因此不完全正确的程序，将可能只得到0分。

4. 仔细阅读题目要求，一定要确保您的输入输出严格符合要求。如输入输出格式不严格遵循题目的要求，会导致被判断为结果错误而不得分。输入输出示例中的注释文字，不是输入输出的组成部分。

5. 提交的源代码必须保持无编译错误，提交有编译错误的程序该题直接得-10分。

6. 除题目另有要求外，程序执行时间应在0.1秒之内，程序中的死循环恕不等待；内存占用不得超过128MB，超过限制的程序将不能通过测试。

7. 所有题目的测试数据不保证数据具有生活常识上的合理性。

8. 不要编写破坏性程序，否则产生的结果对您也是破坏性的，即取消评分资格。

第一题**:** 二分查找

问题定义

大家一定都能熟练掌握二分查找啦！那么来计算二分的次数吧！

约定二分的中点mid = (left + right) / 2。

输入：

第一行输入一个整数N（N<=10000）。

第二行输入N个升序整数。

第三行输入一个待查找的整数（必定在第二行中出现过）。

输出：

输出二分查找该整数时，进行过多少次二分。

输入样例

5

18 53 54 74 99

53

输出样例

2

第二题**:** 计算两个字符串的编辑距离

问题定义

把两个字符串变成相同的三个基本操作定义如下：

1. 修改一个字符（如把a 变成b）

2. 增加一个字符(如abed 变成abedd)

3. 删除一个字符（如jackbllog 变成jackblog）

针对于jackbllog 到jackblog 只需要删除一个或增加一个l 就可以把两个字符串变为相同。

把这种操作需要的最小次数定义为两个字符串的编辑距离L。

编写程序计算指定文件中字符串的距离。输入两个长度不超过512 字节的ASCII 字符串，在

屏幕上输出字符串的编辑距离。

输入样例

Hello world!

Hello word!

输出样例

13

第三题：二叉树遍历

问题定义

输入一棵二叉树，输出树的前、中、后序遍历结果。

输入一个整数N（N<= 10000)，表示树中有N个结点（编号0~N-1）。

接下来N行，依次为结点0~结点N-1的左右孩子情况。

每行3个整数，F,L,R。L,R为F的左右孩子。L,R如果为-1表示该位置上没有孩子。

分三行分别输出树的前中后序遍历。

同一行中的数字，用一个空格间隔。

输入样例

5

0 3 1

1 2 -1

2 -1 4

3 -1 -1

4 -1 -1

输出样例

0 3 1 2 4

3 0 2 4 1

3 4 2 1 0

第四题：**Hanoi** 塔

问题定义

Hanoi 塔问题是印度的一个古老的传说。开天辟地的神勃拉玛在一个庙里留下了三根金刚石的棒，第一根上面套着64 个圆的金片，最大的一个在底下，其余一个比一个小，依次叠上去，庙里的众僧不倦地把它们一个个地从这根棒搬到另一根棒上，规定可利用中间的一根棒作为帮助，但每次只能搬一个，而且大的不能放在小的上面。

请编写程序，把A 柱上的n 个金片，搬动到C 柱（中间可以使用B 柱），使得搬动的次数最少。输入金片的个数n（1<=n<=64），输出总搬动次数，以及最后100 次搬动。如果搬动次数小于等于100 则全部输出；每个搬动占一行，加上是这第几次搬动的数字和”:”，格式见示例。

输入样例

2

输出样例

3

1:A->B

2:A->C

3:B->C