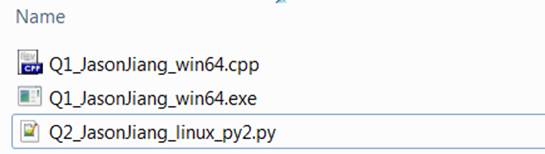
**比赛规则：**

1. 比赛开始时间：4/17周一早晨9:00；比赛结束时间：4/27周四早晨10:00。
2. 请在截止时间之前将答案发送给: [Jason.Jiang@alcatel-lucent.com，Bin.G.Yang@alcatel-lucent.com](mailto:Jason.Jiang@alcatel-lucent.com，Bin.G.Yang@alcatel-lucent.com)。发送答案超过限时者不予考虑。同分情况下，尽早提交答案者排名靠前。允许多次提交答案，以最后一次答案为准。
3. **4月份Code Jam有两道程序题，第一题为基础题100分，第二题为附加题20分**。
4. 提交文件要求：c/c++ 源文件+可执行文件；python/java 源文件。试题答案通过自动化用例评分，将会随机查看部分源代码，基本相似的两份源程序均视为无效答案。

最终提交文件为压缩包，命名规则EnglishName\_languege.tar，如JasonJiang\_Java.tar

压缩包内文件命名规则：questionNumber\_EnglishName\_os.suffix，如



注：Python需注明py2还是py3.

1. C/C++ 程序的编译运行方式：将会用 g++ -O2 进行编译，(version g++ 4.x)
2. Java 程序的运行方式：javac  HelloJava.java ; java HelloJava
3. 程序执行限时：如果题目没有特殊要求，程序运行时间限时1秒 (C/C++/Java) , 15秒(python). 如果程序执行测试例超时，则该测试例不得分。
4. 程序的所有输入通过标准输入获得，运行结果通过标准输出打印。
5. 不得输出任何题目没有要求的输出信息（比如log, 提示信息 "input:" "Succeed!" 等等）。输出多余信息会影响case自动执行通过率，从而影响成绩。
6. 每个test case全对算对，没有半对的说法。Case超时算错。

评分依据：算法正确性、性能、健壮性、代码质量、提交时间。

如有疑问请与各组coordinator联系

BSC BST – Jason Jiang

BSC OAM – Kerry Gao

BSC PF – Kevin Ma

BSC TEL/PM – Felix Zhang

BSC TPHW – Crystal Shan

L4 – Shirley Zhao/John Zhang

MFS – Belinda Zhao

ST – Sarah Zhao

**第一题：**盲人过独木桥

时间限制：1.0s

【问题描述】

长100米的独木桥上有n个盲人。他们有的面向左侧，有的面向右侧。

每个盲人都只能沿着独木桥向前走，速度是1米/秒。

当两个盲人相遇时，他们会同时掉头往回走。

在这些盲人当中，有1个盲人患了传染病。这种传染病相当厉害，以至于在和其他盲人相遇时，一定会会把这种病传染给碰到的盲人。

请你计算，当所有盲人都离开独木桥时，有几个盲人患上了这种传染病。

【输入格式】

第一行输入一个整数n (1 < n < 50), 表示盲人的数量。

接着的一行是n个用空格分开的整数 Xi (-100 < Xi < 100), Xi的绝对值，表示盲人距离独木桥左边端点的距离。正值表示面向右，负值表示面向左，数据中不会出现0值，也不会出现两个盲人占用同一位置。其中，第一个数据代表的盲人患有传染病。

【输出格式】

　　要求输出1个整数，表示最后患上传染病的盲人的数量。

【友情提示】注意排除干扰条件；不考虑异常输入.

【样例输入】

3

5 -2 8

【样例输出】

1

【样例输入】

5

-10 8 -20 12 25

【样例输出】

3

**第二题：**最大利润

时间限制：1.0s

【问题描述】

某商店现有N（2<=N<=11）件商品即将到期，准备两两捆绑销售，若有剩余的单个商品将不再出售，而作为赠品送出。现将N件商品由1到N编号，并给出任意两件商品i和j捆绑销售的利润为Pij (1<=i<=N,1<=j<=N,0<Pij<=100)，请设计算法使得这N件商品获得利润最大。

【输入格式】

第一行输入一个整数N（2<=N<=11）表示商品件数；

接下来的N行，输入了一个N行N列的矩阵，第i行第j列的数字Pij就表示商品i和j组合后的利润，这里我们认为Pij=Pji，Pii=0；

【输出格式】

要求输出1个整数，表示这N件商品的最大利润。

【友情提示】不考虑异常输入.

【样例输入】

3

0 6 62

6 0 35

62 35 0

【样例输出】

62

【样例输入】

4

0 6 62 13

6 0 35 94

62 35 0 5

13 94 5 0

【样例输出】

156