2018年东莞市信息学特长生测试题

**题目概览**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 中文题目名称 | **密码问题** | **括号匹配问题** | **建学校问题** | **平板游戏问题** |
| 题目名称 | pass | bracket | school | platforme |
| 提交文件名 | pass.pas或  pass.cpp | bracket.pas或bracket.cpp | school.pas或school.cpp | platforme.pas或  platforme.cpp |
| 输入文件名 | pass.in | bracket.in | school.in | platforme.in |
| 输出文件名 | pass.out | bracket.out | school.out | platforme.out |
| 测试点时限 | 1秒 | 1秒 | 1秒 | 1秒 |
| 测试点数目 | 10 | 10 | 10 | 10 |
| 测试点分值 | 10 | 10 | 10 | 10 |
| 内存限制 | 256M | 256M | 256M | 256M |

**注意事项：**

1、请严格按题目要求，包括提交文件名、输入输出文件名，否则将严重影响得分。

2、测评时用cena自动测评系统测评，只测pas或 c或 c++源程序。

**一、密码问题**

【问题描述】

ABC公司的大门有一把密码锁，以前锁的密码是一个固定的四位数，为了安全起见，公司对密码锁进行了升级，现在开密码锁先要按一个N键，然后密码锁会随机产生一个四位的正整数，密码是这个值dc 四位正整数的数字构成的最大数与最小数的差值。现在告诉你密码锁随机产生的四位，请你求出所产生的密码。

【数据输入】

只有一个四位正整数，表示密码锁随机产生的四位数。

【数据输出】

只有一个数，表示该密码锁对应的密码。

【输入输出样例1】的

pass.in

4386

pass.out

5175

注：8643-3468=5175

【输入输出样例2】

pass.in

3333

pass.out

0

注：3333-3333=0

**二、括号匹配问题**

【问题描述】

　　一个只由大括号{}，中括号[]，小括号（）组成的字符串，其嵌套次序只能是：大括号可以嵌套中小括号，中括号可以嵌套小括号。现给出n个这样的字符串（每行一个字符串，字符串中无空格），判断其是否正确匹配，若匹配，该行输出“YES”，否则该行输出“NO”（YES和NO均为大写字符，输出时不要输出引号）。

如下是正确匹配样例：

{[()]()[()]}

()()[()]{[()]}

()[()]

{()()[]}

以下不匹配样例：

（[]） 注：中括号置于小括号中

[()(] 注：括号不成对

【数据输入】

　　第一行，n的值（小于10）。

　　接下来n行，每行一个字符串，每个字符串的长度均不超过250个字符。

【数据输出】

　　共n行，每行为“YES”或“NO”，以判断对应行是否匹配。

【输入输出样例】

bracket.in

6 //n为6,表示有6个字符串待检测

{[() ]()[() ]} //以下为6个字符串，每行一个

()() [()]{[()]}

([])

[(()(]

()[()]

{()()[ ] }

bracket.out

YES

YES

NO

NO

YES

YES

**三、建学校问题**

【问题描述】

随着二孩政策的实施，适龄儿童越来越多，学校学位越来越紧张，于是学校的建设工作被放在了重要的议事日程上：

在一条新建的公路两旁，有n个村庄，编号为１,２,…，n。（３≤n≤100），每个村庄有一定数量的小学生，村庄之间的距离也已知。例如n=3时，下图给出了三个村庄的相关情况：

村庄１的学生数２４, 村庄２的学生数１８, 村庄３的学生数３１.

村庄１和村庄２之间距离为１０, 村庄２和村庄３之间的距离为８。

　　　　 １０　　　　8

①————②————③

　　　２４　　　１８　　　３１

现在要在村庄中建k个学校（１≤k≤10）,比如上图中k=2时，建二个学校，此时有三种方案：

村庄１、村庄２，村庄３学生走到村庄２, 学生走的距离和为31×8＝248；

村庄1、村庄3，村庄2学生走到村庄3（就近入学），学生走的距离和为18×8＝144；

村庄２、村庄３，此时村庄１的学生走到村庄２的距离和为24×10＝240

显然，方案２最好。

　　当给出村庄数n，每个村庄的学生数、村庄之间的距离以及要建的学校数k之后，求出合理的建学校的方案，使学生走的距离和为最少。

【数据输入】

第一行２个整数，表示村庄数n，学校数k

第二行n个整数，表示每个村庄学生数（０＜每个村庄学生数≤100 )

第三行n-1个数，分别表示村庄i到村庄i+1之间距离（１≤距离≤100，１≤i≤n-1）

【数据输出】

　　一个整数，即学生走的距离和的最小值

【输入输出样例】

School.in

3 2

21 18 31

10 8

School.out

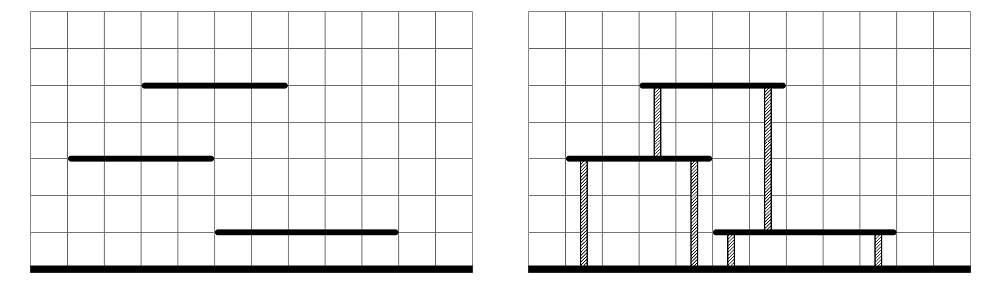
144

**四、平板游戏问题**

【问题描述】

为了进行一种游戏，现决定搭造一些平板，而各个平板的地址已经选定。基于最普遍的认识，没有任何支持物的平板不可能漂浮在空中。说的更精确些，任意一平板的两端必需有支柱或者它在另一块平板上。

你会得到各个平板在坐标系中的坐标（如左下图）。每一块平板的坐标都是由它的高度（与地板间的垂直距离）和它的水平方位（开始和结束）决定的。每个支柱都距它支撑的平板的边缘半个单位（如右下图）。



3个平板，最矮的平板高1，其次的高3，最高的高5。

所需支柱的总长度为14。

算出支持所有平板的支柱的总长度。

【数据输入】

第一行包括1个整数N，1 ≤ N ≤ 100，即平板的总数。

接下来的N行每行都是一块平板的坐标，是相应的Y，X1和 X2。即高度和水平的边缘坐标。所有的数都是不大于10000的正整数且满足X2 > X1+1（也可这样理解，每一块平板的长度至少为2）。

输入保证任意两块平板间没有重叠部分。

【数据输出】

输出要撑起所有平板所需的支柱的总长度。

【输入输出样例1】

platforme.in

3

1 5 10

3 1 5

5 3 7

platform.out

14

【输入输出样例2】

platforme.in

5

50 50 90

40 40 80

30 30 70

20 20 60

10 10 50

platform.out

200