

一. 题目概况

中文题目名称	运动会	差值	取数游戏	传递游戏
英文题目与子目录名	match	diff	game	pass
可执行文件名	match	diff	game	pass
输入文件名	match.in	diff.in	game.in	pass.in
输出文件名	match.out	diff.out	game.out	pass.out
每个测试点时限	1 秒	1 秒	1 秒	1 秒
测试点数目	10	10	10	20
每个测试点分值	10	10	10	5
附加样例文件	有	有	有	有
结果比较方式	全文比较（过滤行末空格及文末回车）			
题目类型	传统	传统	传统	传统

二. 提交源程序文件名

对于 pascal 语言	match.pas	diff.pas	game.pas	pass.pas
对于 c 语言	match.c	diff.c	game.c	pass.c
对于 c++ 语言	match.cpp	diff.cpp	game.cpp	pass.cpp

三. 编译命令（不包含任何优化开关）

对于 pascal 语言	fpc match.pas	fpc diff.pas	fpc game.pas	fpc pass.pas
对于 c 语言	gcc -o match match.c -lm	gcc -o diff diff.c -lm	gcc -o game game.c -lm	gcc -o pass pass.c -lm
对于 c++ 语言	g++ -o match match.cpp -lm	g++ -o diff diff.cpp -lm	g++ -o game game.cpp -lm	g++ -o pass pass.cpp -lm

四. 运行内存限制

内存上限	128M	128M	128M	128M
------	------	------	------	------

注意事项:

- 1、文件名（程序名和输入输出文件名）必须使用英文小写。
- 2、C/C++中函数 main() 的返回值类型必须是 int，程序正常结束时的返回值必须是 0。
- 3、统一评测时采用的机器配置为：CPU P4 3.0GHz，内存 2G，上述时限以此配置为准。
- 4、特别提醒：评测在 Windows 下进行，评测软件为 cena8.0。

运动会

(match.pas/c/cpp)

【问题描述】

梦幻城市每年为全市初中生兴办一次运动会。为促进各校同学之间的交流，采用特别的分队方式：每一个学校的同学，必须被均匀分散到各队，使得每一队中该校的人数皆相同。为增加比赛的竞争性，希望分成越多队越好。你的任务是给定参赛的学校数以及各校的人数，确定最多可以分成的队数。

【输入格式】

第一行一个正整数 N ，代表学校的个数。

接下来 N 行，每行一个正整数，分别代表这 N 个学校的人数。

【输出格式】

输出共一行一个正整数，为最多可分成的队数。

【输入样例 1】

```
3
12
16
20
```

【输出样例 1】

```
4
```

【输入样例 2】

```
4
400
200
150
625
```

【输出样例 2】

```
25
```

【数据规模】

对于 100% 的数据： $1 \leq N \leq 500$ ；每个学校的人数最多为 10000

差值

(diff.pas/c/cpp)

【问题描述】

有一个长度为 N 的序列 A ，现求使其 $(A_1 - A_2) + (A_2 - A_3) + \dots + (A_{N-1} - A_N)$ 结果最大的一种排列，若有多种排列满足此条件，输出字典序最小的那种。

序列 B 的字典序比序列 C 小，当且仅当存在 $i (i < N)$ 满足 $B_i < C_i$ ，且对于每个 $j (0 < j < i)$ 都有 $B_j = C_j$ 。【输入格式】

第一行一个整数 N ，表示序列的长度。

第二行 N 个整数表示序列 A ，第 i 个整数为 A_i ，每两个整数之间用一个空格隔开。

【输出格式】

输出共一行 N 个整数，即所求的排列，每两个整数之间用一个空格隔开，行末不要有多余空格。

【输入样例】

```
5
100 -100 50 0 -50
```

【输出样例】

```
100 -50 0 50 -100
```

【数据规模】

- 对于 30% 的数据： $1 \leq N \leq 10$ ；
- 对于 60% 的数据： $1 \leq N \leq 10000$ ；
- 对于 100% 的数据： $1 \leq N \leq 100000$ ； $|A_i| \leq 1000000$ ；

取数游戏

(game.pas/c/cpp)

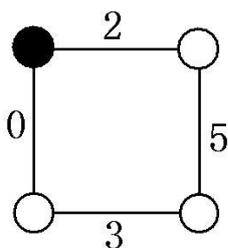
【问题描述】

有一个取数的游戏。初始时，给出一个环，环上的每条边上都有一个非负整数。这些整数中至少有一个 0。然后，将一枚硬币放在环上的一个节点上。两个玩家就是以这个放硬币的节点为起点开始这个游戏，两人轮流取数，取数的规则如下：

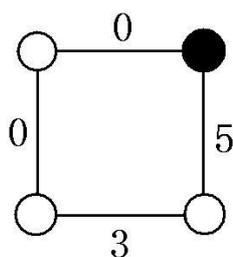
- (1) 选择硬币左边或者右边的一条边，并且边上的数非 0；
- (2) 将这条边上的数减至任意一个非负整数(至少要有所减小)；
- (3) 将硬币移至边的另一端。

如果轮到一个人走，这时硬币左右两边的边上的数值都是 0，那么这个玩家就输了。

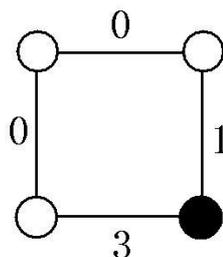
如下图，描述的是 Alice 和 Bob 两人的对弈过程，其中黑色节点表示硬币所在节点。结果图(d)中，轮到 Bob 走时，硬币两边的边上都是 0，所以 Alice 获胜。



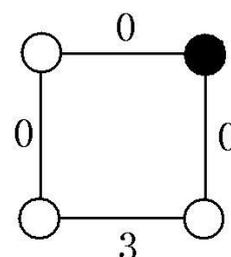
(a) Alice



(b) Bob



(c) Alice



(d) Bob

现在，你的任务就是根据给出的环边上的数值以及起点（硬币所在位置），判断先走一方是否有必胜的策略。

【输入格式】

第一行为测试的组数 T ；对于每组数据格式如下：

第一行一个整数 N ，表示环上的节点数；

第二行 N 个数，第 i 个数 A_i 表示第 i 条边上的数。硬币的起始位置在第 1 条边与第 N 条边之间的节点上。

【输出格式】

输出共 T 行，若先走一方有必胜策略，则输出“Yes”，否则输出“No”；注意大小写，不包含引号。

【输入样例 1】

1
4
2 5 3 0

【输出样例 1】

Yes

【输入样例 2】

2
3
0 0 0
4
2 5 3 0

【输出样例 2】

No
Yes

【数据规模】

对于 30%的数据: $1 \leq T \leq 5$; $2 \leq N \leq 10$;
对于 60%的数据: $1 \leq T \leq 10$; $2 \leq N \leq 50$;
对于 100%的数据: $1 \leq T \leq 100$; $2 \leq N \leq 100$; $0 \leq A_i \leq 100$;

传递游戏

(pass.pas/c/cpp)

【问题描述】

毛大神最近在玩一个传递游戏，即有 N 个人在做传递物品的游戏，这 N 个人的编号为 $1, 2, 3, \dots, N-1, N$ 。游戏规则是这样的：开始时物品可以在任意一人手上，他可将物品传递给其他人中的任意一位；下一个人可以传递给未接过物品的任意一人。

即物品只能经过同一个人一次，而且每次传递过程都有一个代价；不同的人传给不同的人的代价之间没有联系；

毛大神想知道当物品经过所有 N 个人后，整个过程的最小的代价总和是多少。

【输入格式】

第一行为 N , 表示共有 N 个人；

以下为 $N \times N$ 的矩阵，第 i 行第 j 列的数为 $A[i][j]$, 表示物品从编号为 i 的人传递到编号为 j 的人所花费的代价，其中 $A[i][i] = -1$ (因为物品不能自己传给自己)。

【输出格式】

输出共一行一个数，为最小的代价总和。

【输入样例】

2
-1 9794
2724 -1

【输出样例】

2724

【数据规模】

对于 50%的数据： $1 \leq N \leq 11$ ；

对于 100%的数据： $1 \leq N \leq 16; 1 \leq A[i][j] \leq 10000; A[i][i] = -1$ ；