

全国信息学奥林匹克联赛（NOIP2010）复赛

普及组

（请选手务必仔细阅读本页内容）

一. 题目概况

中文题目名称	数字统计	接水问题	导弹拦截	三国游戏
英文题目与子目录名	two	water	missile	sanguo
可执行文件名	two	water	missile	sanguo
输入文件名	two.in	water.in	missile.in	sanguo.in
输出文件名	two.out	water.out	missile.out	sanguo.out
每个测试点时限	1 秒	1 秒	1 秒	1 秒
测试点数目	10	10	10	10
每个测试点分值	10	10	10	10
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	two.pas	water.pas	missile.pas	sanguo.pas
对于 C 语言	two.c	water.c	missile.c	sanguo.c
对于 C++语言	two.cpp	water.cpp	missile.cpp	sanguo.cpp

三. 编译命令（不包含任何优化开关）

对于 pascal 语言	fpc two.pas	fpc water.pas	fpc missile.pas	fpc sanguo.pas
对于 C 语言	gcc -o two two.c -lm	gcc -o water water.c -lm	gcc -o missile missile.c -lm	gcc -o sanguo sanguo.c -lm
对于 C++语言	g++ -o two two.cpp -lm	g++ -o water water.cpp -lm	g++ -o missile missile.cpp -lm	g++ -o sanguo sanguo.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、全国统一评测时采用的机器配置为：CPU P4 3.0GHz，内存 1G，上述时限以此配置为准。
各省在自测时可根据具体配置调整时限。

1. 数字统计

(two.pas/c/cpp)

【问题描述】

请统计某个给定范围[L, R]的所有整数中，数字 2 出现的次数。

比如给定范围[2, 22]，数字 2 在数 2 中出现了 1 次，在数 12 中出现 1 次，在数 20 中出现 1 次，在数 21 中出现 1 次，在数 22 中出现 2 次，所以数字 2 在该范围内一共出现了 6 次。

【输入】

输入文件名为 two.in。

输入共 1 行，为两个正整数 L 和 R，之间用一个空格隔开。

【输出】

输出文件名为 two.out。

输出共 1 行，表示数字 2 出现的次数。

【输入输出样例 1】

two.in	two.out
2 22	6

【输入输出样例 2】

two.in	two.out
2 100	20

【数据范围】

$1 \leq L \leq R \leq 10000$ 。

2. 接水问题

(water.pas/c/cpp)

【问题描述】

学校里有一个水房，水房里一共装有 m 个龙头可供同学们打开水，每个龙头每秒钟的供水量相等，均为 1。

现在有 n 名同学准备接水，他们的初始接水顺序已经确定。将这些同学按接水顺序从 1 到 n 编号， i 号同学的接水量为 w_i 。接水开始时，1 到 m 号同学各占一个水龙头，并同时打开水龙头接水。当其中某名同学 j 完成其接水量要求 w_j 后，下一名排队等候接水的同学 k 马上接替 j 同学的位置开始接水。这个换人的过程是瞬间完成的，且没有任何水的浪费。即 j 同学第 x 秒结束时完成接水，则 k 同学第 $x+1$ 秒立刻开始接水。若当前接水人数 n' 不足 m ，则只有 n' 个龙头供水，其它 $m-n'$ 个龙头关闭。

现在给出 n 名同学的接水量，按照上述接水规则，问所有同学都接完水需要多少秒。

【输入】

输入文件名为 `water.in`。

第 1 行 2 个整数 n 和 m ，用一个空格隔开，分别表示接水人数和龙头个数。

第 2 行 n 个整数 w_1 、 w_2 、……、 w_n ，每两个整数之间用一个空格隔开， w_i 表示 i 号同学的接水量。

【输出】

输出文件名为 `water.out`。

输出只有一行，1 个整数，表示接水所需的总时间。

【输入输出样例 1】

<code>water.in</code>	<code>water.out</code>
5 3 4 4 1 2 1	4

【输入输出样例 1 说明】

第 1 秒，3 人接水。第 1 秒结束时，1、2、3 号同学每人的已接水量为 1，3 号同学接完水，4 号同学接替 3 号同学开始接水。

第 2 秒，3 人接水。第 2 秒结束时，1、2 号同学每人的已接水量为 2，4 号同学的已接水量为 1。

第 3 秒，3 人接水。第 3 秒结束时，1、2 号同学每人的已接水量为 3，4 号同学的已接水量为 2。4 号同学接完水，5 号同学接替 4 号同学开始接水。

第 4 秒，3 人接水。第 4 秒结束时，1、2 号同学每人的已接水量为 4，5 号同学的已接水量为 1。1、2、5 号同学接完水，即所有人完成接水。

总接水时间为 4 秒。

【输入输出样例 2】

<code>water.in</code>	<code>water.out</code>
8 4 23 71 87 32 70 93 80 76	163

【数据范围】

$1 \leq n \leq 10000$ ， $1 \leq m \leq 100$ 且 $m \leq n$ ；

$1 \leq w_i \leq 100$ 。

3. 导弹拦截

(`missile.pas/c/cpp`)

【问题描述】

经过 11 年的韬光养晦，某国研发出了一种新的导弹拦截系统，凡是与它的距离不超过其工作半径的导弹都能够被它成功拦截。当工作半径为 0 时，则能够拦截与它位置恰好相同的导弹。但该导弹拦截系统也存在这样的缺陷：每套系统每天只能设定一次工作半径。而当

天的使用代价，就是所有系统工作半径的平方和。

某天，雷达捕捉到敌国的导弹来袭。由于该系统尚处于试验阶段，所以只有两套系统投入工作。如果现在的要求是拦截所有的导弹，请计算这一天的最小使用代价。

【输入】

输入文件名 missile.in。

第一行包含 4 个整数 x_1 、 y_1 、 x_2 、 y_2 ，每两个整数之间用一个空格隔开，表示这两套导弹拦截系统的坐标分别为 (x_1, y_1) 、 (x_2, y_2) 。

第二行包含 1 个整数 N ，表示有 N 颗导弹。接下来 N 行，每行两个整数 x 、 y ，中间用一个空格隔开，表示一颗导弹的坐标 (x, y) 。不同导弹的坐标可能相同。

【输出】

输出文件名 missile.out。

输出只有一行，包含一个整数，即当天的最小使用代价。

【提示】

两个点 (x_1, y_1) 、 (x_2, y_2) 之间距离的平方是 $(x_1 - x_2)^2 + (y_1 - y_2)^2$ 。

两套系统工作半径 r_1 、 r_2 的平方和，是指 r_1 、 r_2 分别取平方后再求和，即 $r_1^2 + r_2^2$ 。

【输入输出样例 1】

missile.in	missile.out
0 0 10 0 2 -3 3 10 0	18

【样例 1 说明】

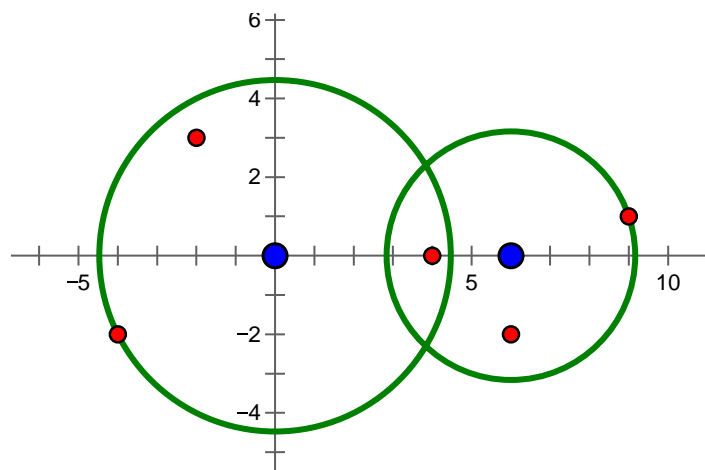
样例 1 中要拦截所有导弹，在满足最小使用代价的前提下，两套系统工作半径的平方分别为 18 和 0。

【输入输出样例 2】

missile.in	missile.out
0 0 6 0 5 -4 -2 -2 3 4 0 6 -2 9 1	30

【样例 2 说明】

样例中的导弹拦截系统和导弹所在的位置如下图所示。要拦截所有导弹，在满足最小使用代价的前提下，两套系统工作半径的平方分别为 20 和 10。

**【数据范围】**

对于 10%的数据， $N = 1$

对于 20%的数据， $1 \leq N \leq 2$

对于 40%的数据， $1 \leq N \leq 100$

对于 70%的数据， $1 \leq N \leq 1000$

对于 100%的数据， $1 \leq N \leq 100000$ ，且所有坐标分量的绝对值都不超过 1000。

4. 三国游戏

(`sanguo.pas/c/cpp`)

【问题描述】

小涵很喜欢电脑游戏，这些天他正在玩一个叫做《三国》的游戏。

在游戏中，小涵和计算机各执一方，组建各自的军队进行对战。游戏中共有 N 位武将 (N 为偶数且不小于 4)，任意两个武将之间有一个“默契值”，表示若此两位武将作为一对组合作战时，该组合的威力有多大。游戏开始前，所有武将都是自由的（称为自由武将，一旦某个自由武将选中作为某方军队的一员，那么他就不再是自由武将了），换句话说，所谓的自由武将不属于任何一方。游戏开始，小涵和计算机要从自由武将中挑选武将组成自己的军队，规则如下：小涵先从自由武将中选出一个加入自己的军队，然后计算机也从自由武将中选出一个加入计算机方的军队。接下来一直按照“小涵→计算机→小涵→……”的顺序选择武将，直到所有的武将被双方均分完。然后，程序自动从双方军队中各挑出一对默契值最高的武将组合代表自己的军队进行二对二比武，拥有更高默契值的一对武将组合获胜，表示两军交战，拥有获胜武将组合的一方获胜。

已知计算机一方选择武将的原则是尽量破坏对手下一步将形成的最强组合，它采取的具体策略如下：任何时刻，轮到计算机挑选时，它会尝试将对手军队中的每个武将与当前每个自由武将进行一一配对，找出所有配对中默契值最高的那对武将组合，并将该组合中的自由武将选入自己的军队。

下面举例说明计算机的选将策略，例如，游戏中一共有 6 个武将，他们相互之间的默契

值如下表所示

武将 编号	1	2	3	4	5	6
1		5	28	16	29	27
2	5		23	3	20	1
3	28	23		8	32	26
4	16	3	8		33	11
5	29	20	32	33		12
6	27	1	26	11	12	

双方选将过程如下所示：

	小涵	轮到计算机时可 选的自由武将	计算机	计算机选将说明
第一轮	5	1 2 3 4 6	4	小涵手中 5 号武将与 4 号的默契值 最高，所以选择 4 号
第二轮	5 3	1 2 6	4 1	小涵手中的 5 号和 3 号武将与自由 武将中配对可产生的最大默契值 为 29，是由 5 号与 1 号配对产生 的，因此计算机选择 1 号
第三轮	5 3 6	2	4 1 2	

小涵想知道，如果计算机在一局游戏中始终坚持上面这个策略，那么自己有没有可能必胜？如果有，在所有可能的胜利结局中，自己那对用于比武的武将组合的默契值最大是多少？

假设整个游戏过程中，对战双方任何时候均能看到自由武将队中的武将和对方军队的武将。为了简化问题，保证对于不同的武将组合，其默契值均不相同。

【输入】

输入文件名为 `sanguo.in`，共 N 行。

第一行为一个偶数 N ，表示武将的个数。

第 2 行到第 N 行里，第 $(i+1)$ 行有 $(N-i)$ 个非负整数，每两个数之间用一个空格隔开，表示 i 号武将和 $i+1, i+2, \dots, N$ 号武将之间的默契值 ($0 \leq \text{默契值} \leq 1,000,000,000$)。

【输出】

输出文件 `sanguo.out` 共 1 或 2 行。

若对于给定的游戏输入，存在可以让小涵获胜的选将顺序，则输出 1，并另起一行输出所有获胜的情况中，小涵最终选出的武将组合的最大默契值。

如果不存在可以让小涵获胜的选将顺序，则输出 0。

【输入输出样例 1】

sanguo.in	sanguo.out
6 5 28 16 29 27 23 3 20 1 8 32 26 33 11 12	1 32

【输入输出样例说明】

首先小涵拿走 5 号武将；计算机发现 5 号武将和剩下武将中的 4 号默契值最高，于是拿走 4 号；小涵接着拿走 3 号；计算机发现 3、5 号武将之一和剩下的武将配对的所有组合中，5 号和 1 号默契值最高，于是拿走 1 号；小涵接着拿走 2 号；计算机最后拿走 6 号。在小涵手里的 2, 3, 5 号武将中，3 号和 5 号配合最好，默契值为 32，而计算机能推出的最好组合为 1 号和 6 号，默契值为 27。结果为小涵胜，并且这个组合是小涵用尽所有方法能取到的最好组合。

【输入输出样例 2】

sanguo.in	sanguo.out
8 42 24 10 29 27 12 58 31 8 16 26 80 6 25 3 36 11 5 33 20 17 13 15 77 9 4 50 19	1 77

【数据范围】

对于 40% 的数据有 $N \leq 10$ 。

对于 70% 的数据有 $N \leq 18$ 。

对于 100% 的数据有 $N \leq 500$ 。