

8.3_信心赛

by dst

(选手可以选择无视本页内容)

一. 题目概况

每个测试点时限	见各题题面
每个测试点分值	100/各题测试点数目
附加样例文件	无
结果比较方式	前 3 题: 全文比较 (忽略行末空格及文末回车) 第 4 题: 逐行比较 (忽略行末空格及文末回车)
题目类型	传统
每个测试点运行内存上限	见各题题面

注意事项:

- 1.测试时使用标准输入输出, 而不是文件输入输出。
- 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.若题目没有特殊说明, 则测试数据中所有在同一行的元素间用一个空格隔开, 无多余空格, 文件末尾有且只有一行换行。

A. 安排妹子

(array.cpp/c/pas, 1s, 256MB)

【问题背景】

众所周知，*jl**b*是个把妹小能手。

【问题描述】

假期一共 n 天，每一天他的妹子 qwq 的心情会有一个阻值 a_i 。在假期，*jl**b*每天会花相同的时间 p 安排 qwq ，而他第 i 天能安排上 qwq 的时间为 $\lceil \frac{p}{a_i} \rceil$ 。*jl**b*当然希望能尽可能地安排上 qwq ，但事实上，如果这个假期*jl**b*安排上 qwq 的总时间超过 t ， qwq 就会感到反感。

所以*jl**b*希望找到一个合适的正整数 p ，使得*jl**b*假期安排上 qwq 的总时间恰好为 t 。

注意： $[x]$ 运算是将实数 x 向下取整，例如 $[3.7] = 3$ 。

【输入】

输入共2行。

第1行包含两个非负整数 n, t 。

第2行包含 n 个正整数，第 i 个正整数表示 a_i 。

【输出】

输出共1行，包含1个正整数 p 。若存在多个满足条件的 p ，则输出其中最小的；若没有满足条件的 p ，则输出"The probability of existence of p is 0%."（其中""不输出）。

【输入输出样例 1】

array.in	array.out
3 5 6 2 5	6

【输入输出样例 2】

array.in	array.out
3 5 4 1 3	The probability of existence of p is 0%.

【数据规模与约定】

对于30%的数据， $n \leq 50; t, a_i \leq 10^3$ 。

对于100%的数据， $1 \leq n \leq 5 \times 10^5; 0 \leq t \leq 10^8; 1 \leq a_i \leq 10^8$ 。

B. 好感度 up

(private.cpp/c/pas, 1s, 256MB)

【问题背景】

自从上了高中，*jl*b与妹子*qw*q独处的时间越来越少，使好感度*up*的机会也越来越少。

【问题描述】

*jl*b决定主动出击。*jl*b和*qw*q现在在高为*l*，长为*m*，宽为*n*的教学楼中。我们用 (i, j, k) 表示教学楼的每个点，用 $a_{i,j,k}$ 描述每个点，0表示空，1表示墙。每一个空点表示一个独处点。

若两个点 (i_1, j_1, k_1) 与 (i_2, j_2, k_2) 满足 $|i_1 - i_2| + |j_1 - j_2| + |k_1 - k_2| = 1$ ，则这两个点相邻。所有相邻的独处点可以作为同一个可以独处的位置。例如在一个高为5，长为5，宽为5的教学楼中，除 $a_{2,2,2} = 0, a_{2,3,2} = 0, a_{2,2,3} = 0$ 外，每个点都是墙，则 $(2,2,2), (2,3,2), (2,2,3)$ 为一个可以独处的位置。

若一个点 (i, j, k) 满足 $i < 1$ 或 $i > l$ 或 $j < 1$ 或 $j > m$ 或 $k < 1$ 或 $k > n$ ，则该点位于外界。如果可以独处的位置中有一点与外界相邻，则整个位置不再算作可以独处的位置。例如在上面的例子中，若 $a_{1,2,2} = 0$ ，则 $(1,2,2), (2,2,2), (2,3,2), (2,2,3)$ 不是一个可以独处的位置，因为 $(1,2,2)$ 与外界相邻。

能够独处的时间越多，*qw*q对*jl*b的好感度就会upper。所以*jl*b想知道他和*qw*q有多少个可以独处的位置。

【输入】

输入共 $(l \times m + 1)$ 行。

第1行包含3个正整数，表示*l, m, n*。

接下来 $l \times m$ 行，共*l*个*m*行，每行包含*n*个值为0或1的整数，其中第*i*个*m*行中的第*j*行的第*k*个数表示 $a_{i,j,k}$ 。

【输出】

输出共1行，1个非负整数，表示*jl*b和*qw*q拥有的可以独处的位置数量。

【输入输出样例】

private.in	private.out
3 4 5 1 1 1 1 1	1

1 1 1 1 1	
1 1 1 1 1	
1 1 1 1 1	
1 1 1 1 1	
1 0 0 1 1	
1 0 1 0 0	
1 1 1 1 1	
1 1 1 1 1	
1 1 1 1 1	
1 1 1 1 1	
1 1 1 1 1	

【数据规模与约定】

对于5%的数据， $l \leq 2$ 或 $m \leq 2$ 或 $n \leq 2$ 。

另外30%的数据， $l, m, n \leq 10$ 。

对于100%的数据， $1 \leq l, m, n \leq 100$ 。

c. 炒鸡矿工

(`chicken.cpp/c/pas`, 1s, 256MB)

【问题背景】

炒鸡矿工不是黄金矿工，而是dst在QQ小游戏中看到的一个沙雕小游戏。



【问题描述】

这个游戏的玩法建立在一个挖矿系统上。

我们认为游戏从第0分钟开始，每过 p 分钟，炒鸡矿工可以完成一次挖矿，每次可以挖重量为 c 的金矿，准确的说，在一次挖矿中，dst会在第 p 分钟末收获重量为 c 的金矿。炒鸡矿工在开局后会不断地重复挖矿操作，不能休息。

金矿可以储存或用于升级挖矿系统。

开局时，挖矿系统的等级为1级。挖矿系统最多升到 $n+1$ 级。升级操作不消耗时间，但只能在一次挖矿开始前进行。每次升级会从第 i 级升级到第 $i+1$ 级($1 \leq i \leq n$)，需要花费重量为 w_i 的金矿，可以使每次挖矿的重量增加 v_i ，使每次挖矿的时间变成 s_i 。由于升级不消耗时间，dst可以在一瞬间多次升级。

开局时，dst拥有重量为 m 的金矿。他想知道，在开局后恰好 t 分钟时，他最多能拥有的金矿重量 g 是多少。

【输入】

输入共4行。

第1行包含5个非负整数 p, c, n, m, t 。

第2行包含 n 个非负整数，第 i 个数表示 w_i 。

第3行包含 n 个非负整数，第 i 个数表示 v_i 。

第4行包含 n 个正整数，第 i 个数表示 s_i 。

若 $n=0$ ，则第2行，第3行，第4行为空行。

【输出】

输出共1行，包含1个非负整数 g 。

【输入输出样例】

chicken.in	chicken.out
3 2 2 1 6	17

1 3	
3 0	
3 1	

【输入输出样例说明】

下面给出一种可行的方案（同一行内相同颜色标记表示相关联的变化）：

	等级	收矿倒计时	拥有金矿重量
开局	1		1
0s	$1 + 1 = 2$	3	$1 - 1 = 0$
1s	2	2	0
2s	2	1	0
3s	$2 + 1 = 3$	$0 + 1 = 1$	$0 + 5 - 3 = 2$
4s	3	$0 + 1 = 1$	$2 + 5 = 7$
5s	3	$0 + 1 = 1$	$7 + 5 = 12$
6s	3	0	$12 + 5 = 17$

【数据规模与约定】

- 对于5%的数据， $n = 0$ 。
- 另外10%的数据， $w_i = 0$ 。
- 另外10%的数据， $v_i = 0$ 。
- 另外10%的数据， $s_i = p$ 。
- 另外10%的数据， $t \leq 100$ 。
- 另外10%的数据， $m, c, w_i, v_i \leq 10$ 。
- 对于80%的数据， $p, s_i \leq 10$ 。
- 对于100%的数据， $1 \leq p, s_i, t \leq 10^3; 0 \leq n \leq 10^3; 0 \leq m, c, w_i, v_i \leq 10^9$ 。

D. 游戏大师

(`master.cpp/c/pas`, 2s, 256MB)

【问题背景】

众所周知, *hk* 是我校的游戏大师(*master*), 尤其擅长疵客信条和英熊联盟。

【问题描述】

现在, 有 n 个 *hk* 要进行角逐, 其中第 i 个 *hk* 对疵客信条的擅长度为 a_i , 对英熊联盟的擅长度为 b_i , 保证每一个 *hk* 对同一个游戏的擅长度不同, 即对于任意 $i, j (i \neq j)$, $a_i \neq a_j$ 且 $b_i \neq b_j$ 。由于 *hk* 是游戏大师, 对自己的游戏水平极端自信, 因此, 对于任意 $i, j (i \neq j)$, 第 i 个 *hk* 认为, 只要自己对任一游戏的擅长度高于第 j 个 *hk*, 即 $a_i > a_j$ 或 $b_i > b_j$, 他就一定能打败第 j 个 *hk*。

然而, *zfqh* 大神(*god*)极端质疑 *hk* 的实力, 因此, 对于任意 $i, j (i \neq j)$, *zfqh* 认为, 只有第 i 个 *hk* 对所有游戏的擅长度不低于第 j 个 *hk* 的两倍, 即 $a_i \geq a_j \times 2$ 且 $b_i \geq b_j \times 2$, 他才一定能打败第 j 个 *hk*。

现在, 作为一名吃瓜群众, *dst* 想知道, 在 *hk* 和 *zfqh* 的观点中, 每一个 *hk* 分别一定能打败多少个其他的 *hk*。

【输入】

输入共3行。

第1行包含1个正整数 n 。

第2行包含 n 个正整数, 第 i 个数表示 a_i 。

第3行包含 n 个正整数, 第 i 个数表示 b_i 。

【输出】

输出共2行。

第1行包含 n 个非负整数, 第 i 个数表示在 *hk* 的观点中, 第 i 个 *hk* 一定能打败多少个其他的 *hk*。

第2行包含 n 个非负整数, 第 i 个数表示在 *zfqh* 的观点中, 第 i 个 *hk* 一定能打败多少个其他的 *hk*。

【输入输出样例】

<code>master.in</code>	<code>master.out</code>
4	0 2 2 3
1 4 3 5	0 1 1 1

1 3 4 5	
---------	--

【数据规模与约定】

本题共100分，10个数据点，输出按行比较给分，每行5分。

对于20%的数据， $n \leq 10^3$ 。

另外20%的数据， $a_i = b_i$ 。

另外40%的数据， $a_i, b_i \leq n$ 。

对于100%的数据， $1 \leq n \leq 10^5; 1 \leq a_i, b_i \leq 10^9$ 。