

最大子矩阵(matrix.cpp)

总时间限制: 1s

内存限制: 64MB

【问题描述】

已知矩阵的大小定义为矩阵中所有元素的和。给定一个矩阵，你的任务是找到最大的非空(大小至少是 $1 * 1$)子矩阵。

比如，如下 $4 * 4$ 的矩阵

```
0  -2  -7  0
9   2  -6  2
-4   1  -4  1
-1   8   0 -2
```

的最大子矩阵是

```
9   2
-4   1
-1   8
```

这个子矩阵的大小是 15。

【输入格式】

输入是一个 $N * N$ 的矩阵。输入的第一行给出 N ($0 < N \leq 100$)。再后面的若干行中，依次（首先从左到右给出第一行的 N 个整数，再从左到右给出第二行的 N 个整数……）给出矩阵中的 N^2 个整数，整数之间由空白字符分隔（空格或者空行）。已知矩阵中整数的范围都在 $[-127, 127]$ 。

【输出格式】

输出最大子矩阵的大小。

【输入样例】(matrix.in)

```
4
0  -2  -7  0
9   2  -6  2
-4   1  -4  1
-1   8   0 -2
```

【输出样例】(matrix.out)

```
15
```

接水问题 (water.cpp)

总时间限制: 1s

内存限制: 64MB

【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。

【输入格式】

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, \dots, 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。 $1 \leq n \leq 10000$, $1 \leq m \leq 100$ 且 $m \leq n$; $1 \leq w_i \leq 100$ 。

【输出格式】

输出只有一行，1 个整数，表示接水所需的总时间。

【输入样例】(water.in)

```
5 3
4 4 1 2 1
```

【输出样例】(water.out)

```
4
```

装箱问题 (packing.cpp)

总时间限制: 1s

内存限制: 64MB

【问题描述】

一个工厂制造的产品形状都是长方体，它们的高度都是 h ，长和宽都相等，一共有六个型号，他们的长宽分别为 $1*1, 2*2, 3*3, 4*4, 5*5, 6*6$ 。这些产品通常使用一个 $6*6*h$ 的长方体包裹包装然后邮寄给客户。因为邮费很贵，所以工厂要想办法的减小每个订单运送时的包裹数量。他们很需要有一个好的程序帮他们解决这个问题从而节省费用。现在这个程序由你来设计。

【输入格式】

输入文件包括几行，每一行代表一个订单。每个订单里的一行包括六个整数，中间用空格隔开，分别为 $1*1$ 至 $6*6$ 这六种产品的数量。输入文件将以 6 个 0 组成的一行结尾。

【输出格式】

除了输入的最后一行 6 个 0 以外，输入文件里每一行对应着输出文件的一行，每一行输出一个整数代表对应的订单所需的最小包裹数。

【输入样例】(packing.in)

```
004001
751000
000000
```

【输出样例】(packing.out)

```
2
1
```

Ride to Office (office.cpp)

总时间限制: 1s

内存限制: 64MB

【问题描述】

许多员工住在一个叫 MZone 的地方，离他们的办公室很远（4.5 公里）。由于交通不便，许多工作人员选择骑自行车。Charley 是一个有不同骑行习惯的人——他总是试图以相同的速度跟随另一个人来避免单独骑行。Charley 会寻找正在出发到办公室的人，如果他找到某人，他会跟随那个人并保持相同的骑行速度，或者如果没有找到，他会等待某人然后跟随。在从 Charley 的家到办公室的路上，任何时候如果一个更快的人超越了 Charley，Charley 就会离开他所追随的人并加速跟上这个更快的车手。你的任务是计算出 Charley 用他独特的骑行方式到达办公室共需要多少时间。结果保留整数（向上取整）。

【输入格式】

若干组数据，每组数据的第一行是 N ($1 \leq N \leq 10000$)，表示骑行人数（不包括 Charley）， $N = 0$ 结束输入。接下来 N 行是 N 个不同车手的信息，格式为: $V_i \ T_i$

V_i 是一个小于等于 40 的正整数，表示第 i 位车手的速度 (kph, 公里/小时)。 T_i 是第 i 位骑手的出发时间，它是一个整数并以秒为单位计算。在任何情况下，总是存在一个非负 T_i 。如果 $T_i < 0$ ，表示某人提早出发了。

【输出格式】

每组数据对应输出一行：即 Charley 的到达时间。

【输入样例】(office.in)

```
4
20 0
25 -155
27 190
30 240
2
21 0
22 34
0
```

【输出样例】(office.out)

```
780
771
```

书架 (bookshelf.cpp)

总时间限制: 1s

内存限制: 64MB

【问题描述】

John 最近买了一个书架用来存放奶牛养殖书籍, 但书架很快被存满了, 只剩最顶层有空余。

John 共有 N 头奶牛($1 \leq N \leq 20,000$), 每头奶牛有自己的高度 H_i ($1 \leq H_i \leq 10,000$), N 头奶牛的总高度为 S 。书架高度为 B ($1 \leq B \leq S < 2,000,000,00$)。

为了到达书架顶层, 奶牛可以踩着其他奶牛的背, 像叠罗汉一样, 直到他们的总高度不低于书架高度。当然若奶牛越多则危险性越大。为了帮助 John 到达书架顶层, 找出使用奶牛数目最少的解决方案吧。

【输入格式】

第 1 行: 空格隔开的整数 N 和 B 。

第 2~ $N+1$ 行: 第 $i+1$ 行为整数 H_i 。

【输出格式】

能达到书架高度所使用奶牛的最少数目。

【输入样例】(bookshelf.in)

```
6 40
6
18
11
13
19
11
```

【输出样例】(bookshelf.out)

```
3
```

电池的使用寿命 (battery.cpp)

总时间限制: 1s

内存限制: 64MB

【问题描述】

小 S 新买了一个掌上游戏机, 这个游戏机由两节 5 号电池供电。为了保证能够长时间玩游戏, 他买了很多 5 号电池, 这些电池的生产商不同, 质量也有差异, 因而使用寿命也有所不同, 有的能使用 5 个小时, 有的可能就只能使用 3 个小时。显然如果他只有两个电池一个能用 5 小时一个能用 3 小时, 那么他只能玩 3 个小时的游戏, 有一个电池剩下的电量无法使用, 但是如果他有更多的电池, 就可以更加充分地利用它们, 比如 he 有三个电池分别能用 3、3、5 小时, 他可以先使用两节能用 3 个小时的电池, 使用半个小时后再把其中一个换成能使用 5 个小时的电池, 两个半小时后再把剩下的一节电池换成刚才换下的电池(那个电池还能用 2.5 个小时), 这样总共就可以使用 5.5 个小时, 没有一点浪费。

现在已知电池的数量和电池能够使用的时间, 请你找一种方案使得使用时间尽可能的长。

【输入格式】

输入包含多组数据, 输入 0 则结束。每组数据包括两行, 第一行是一个整数 N ($2 \leq N \leq 1000$), 表示电池的数目, 接下来一行是 N 个正整数表示电池能使用的时间。

【输出格式】

对每组数据输出一行, 表示电池能使用的时间, 保留到小数点后 1 位。

【输入样例】(battery.in)

```
2
3 5
3
3 3 5
```

【输出样例】(battery.out)

```
3.0
5.5
```

寻找平面上的极大点 (point.cpp)

总时间限制: 1s

内存限制: 64MB

【问题描述】

在一个平面上,如果有两个点 $(x,y),(a,b)$,如果说 (x,y) 支配了 (a,b) ,这是指 $x \geq a, y \geq b$;用图形来看就是 (a,b) 坐落在以 (x,y) 为右上角的一个无限的区域内。

给定 n 个点的集合,一定存在若干个点,它们不会被集合中的任何一点所支配,这些点叫做极大值点。

编程找出所有的极大点,按照 x 坐标由小到大,输出极大点的坐标。

本题规定: n 不超过 100, 并且不考虑点的坐标为负数的情况。

【输入格式】

输入包括两行,第一行是正整数 n ,表示是点数,第二行包含 n 个点的坐标,坐标值都是整数,坐标范围从 0 到 100,输入数据中不存在坐标相同的点。

【输出格式】

按 x 轴坐标最小到大的顺序输出所有极大点。

输出格式为: $(x_1,y_1),(x_2,y_2),\dots,(x_k,y_k)$

注意:输出的每个点之间有","分隔,最后一个点之后没有",", 少输出和多输出都会被判错。

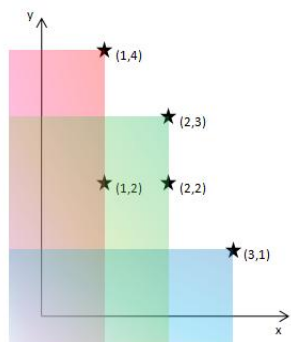
【输入样例】(point.in)

```
5
1 2 2 3 1 2 3 1 4
```

【输出样例】(point.out)

```
(1,4),(2,3),(3,1)
```

【提示】



过河 (river.cpp)

总时间限制: 1s

内存限制: 64MB

【问题描述】

N 个人过河,但只有一条最多可载两人的船。因此,必须安排某种坐船顺序以便来回划船,让所有人都可以过河。每个人都有不同的划船速度,船的速度取决于较慢的划船速度。你的任务是设计一个策略,尽量减少所有人过河的时间。

【输入格式】

输入的第一行包含一个整数 T ($1 \leq T \leq 20$)。即有 T 组测试数据。每组测试数据的第一行输入 N ,第二行输入 N 个整数,表示每个人过河的时间。不会超过 1000 人,没有人需要超过 100 秒才能过河。

【输出格式】

对于每组测试数据,输出一行,表示所有 N 个人过河所需的最少时间。

【输入样例】(river.in)

```
1
4
1 2 5 10
```

【输出样例】(river.out)

```
17
```