排队接水 (order.cpp)

总时间限制: 1s　　内存限制: 64MB

【问题描述】

有 n(n<=1000)个人在一个水龙头前排队接水，假如每个人接水的时间为 T_i ，请编程找出这 n 个人排队的一种顺序，使得 n 个人的平均等待时间最小。

【输入格式】

输入文件共两行，第一行为 n；第二行分别表示第 1 个人到第 n 个人每人的接水时间 T_1 ， T_2 ，…， T_n ，每个数据之间有 1 个空格。

【输出格式】

输出文件有一行，为这种排列方案下的平均等待时间(输出结果精确到小数点后两位)。

【输入样例】（order.in）

```
10
56  12  1  99  1000  234  33  55  99  812
```

【输出样例】（order.out）

```
532.00
```

最大整数(maxnum.cpp）

总时间限制: 1s　　内存限制: 64MB

【问题描述】

设有 n 个正整数（n≤20），将它们联接成一排，组成一个最大的多位整数。

例如：n=3 时，3 个整数 13，312，343 联接成的最大整数为：34331213。

又如：n=4 时，4 个整数 7，13，4，246 联接成的最大整数为：7424613。

【输入格式】

第一行输入 n 。

第二行输入 n 个整数。

【输出格式】

联接成的最大整数。

【输入样例】（maxnum.in）

```
3
13  312  343
```

【输出样例】（maxnum.out）

```
34331213
```

均分纸牌(playcard.cpp）

总时间限制: 1s　　内存限制: 64MB

【问题描述】

有 N 堆纸牌，编号分别为 1，2，…, N。每堆上有若干张，但纸牌总数必为 N 的倍数。可以在任一堆上取若干张纸牌，然后移动。

移牌规则为：在编号为 1 堆上取的纸牌，只能移到编号为 2 的堆上；在编号为 N 的堆上取的纸牌，只能移到编号为 N-1 的堆上；其他堆上取的纸牌，可以移到相邻左边或右边的堆上。现在要求找出一种移动方法，用最少的移动次数使每堆上纸牌数都一样多。

例如 N=4，4 堆纸牌数分别为： ① 9 ② 8 ③ 17 ④ 6

移动 3 次可达到目的：从 ③ 取 4 张牌放到④（9 8 13 10）->从③取 3 张牌放到 ②（9 11 10 10）-> 从②取 1 张牌放到①（10 10 10 10）。

【输入格式】

第一行输入一个整数 N（N 堆纸牌，1 <= N <= 100）；

第二行输入 N 个整数 A_1 、A_2 … A_n （N 堆纸牌，每堆纸牌初始数，1<= A_i <=10000）。

【输出格式】

所有堆均达到相等时的最少移动次数。

【输入样例】（playcard.in）

```
4
9  8  17  6
```

【输出样例】（playcard.out）

```
3
```

删数问题 (delete.cpp）

总时间限制: 1s　　内存限制: 64MB

【问题描述】

输入一个高精度的正整数 N，去掉其中任意 S 个数字后剩下的数字按原左右次序组成一个新的正整数。编程对给定的 N 和 S，寻找一种方案使得剩下的数字组成的新数最小。

输出新的正整数。（N 不超过 240 位）

输入数据均不需判错。

【输入格式】

N

S

【输出格式】

最后剩下的最小数。

【输入样例】（delete.in）

```
175438
4
```

【输出样例】（delete.out）

```
13
```

拦截导弹问题(missile.cpp）

总时间限制: 1s　　内存限制: 64MB

【问题描述】

某国为了防御敌国的导弹袭击，开发出一种导弹拦截系统，但是这种拦截系统有一个缺陷：虽然它的第一发炮弹能够到达任意的高度，但是以后每一发炮弹都不能高于前一发的高度。某天，雷达捕捉到敌国的导弹来袭，由于该系统还在试用阶段。所以一套系统有可能不能拦截所有的导弹。

输入导弹依次飞来的高度（雷达给出的高度不大于 30000 的正整数）。计算要拦截所有导弹最小需要配备多少套这种导弹拦截系统。。

【输入格式】

n 颗依次飞来的导弹高度（1≤n≤1000）。

【输出格式】

要拦截所有导弹最小配备的系统数 k。

【输入样例】（missile.in）

389　207　155　300　299　170　158　65

【输出样例】（missile.out）

2

【输入 输出 样例 说明 】

输入：导弹高度： 7 9 6 8 5

输出：导弹拦截系统 K=2

输入：导弹高度： 4 3 2

输出：导弹拦截系统 K=1

开会问题(meeting.cpp）

总时间限制: 1s　　内存限制: 64MB

【问题描述】

CCS 公司的会议日益增多，以至于全公司唯一的会议室都不够用了。现在给出这段时间的会议时间表（会议之间在时间上可能冲突），要求你适当删除一些会议，使得剩余的会议在时间上互不冲突。要求删除的会议最少。

【输入格式】

第一行是整数 N（N<=500），表示总共的会议数目；

接下来的 N 行，每行两个整数，表示该会议的起止时间。

【输出格式】

文件只有一行，含一个整数，表示删除的最少会议数。

【输入样例】（meeting.in）

3

1　3

3　5

2　5

【输出样例】（meeting.out）

1

合并果子 (fruit.cpp)

总时间限制: 1s　　内存限制: 64MB

【问题描述】

在一个果园里，多多已经将所有的果子打了下来，而且按果子的不同种类分成了不同的堆。多多决定把所有的果子合成一堆。

每一次合并，多多可以把两堆果子合并到一起，消耗的体力等于两堆果子的重量之和。可以看出，所有的果子经过 n-1 次合并之后，就只剩下一堆了。多多在合并果子时总共消耗的体力等于每次合并所耗体力之和。

因为还要花大力气把这些果子搬回家，所以多多在合并果子时要尽可能地节省体力。假定每个果子重量都为 1，并且已知果子的种类数和每种果子的数目，你的任务是设计出合并的次序方案，使多多耗费的体力最少，并输出这个最小的体力耗费值。

例如有 3 种果子，数目依次为 1，2，9。可以先将 1、2 堆合并，新堆数目为 3，耗费体力为 3。接着，将新堆与原先的第三堆合并，又得到新的堆，数目为 12，耗费体力为 12。所以多多总共耗费体力=3+12=15。

可以证明 15 为最小的体力耗费值。

【输入格式】

输入文件 fruit.in 包括两行，第一行是一个整数 n（1 <= n <= 10000），表示果子的种类数。

第二行包含 n 个整数，用空格分隔，第 i 个整数 a_i （1 <= a_i <= 20000）是第 i 种果子的数目。

【输出格式】

输出文件 fruit.out 包括一行，这一行只包含一个整数，也就是最小的体力耗费值。输入数据保证这个值小于 2^{31}。

【输入样例】（fruit.in）

```
3
1 2 9
```

【输出样例】（fruit.out）

```
15
```

【数据规模】

对于 30%的数据，保证有 n <= 1000；

对于 50%的数据，保证有 n <= 5000；

对于全部的数据，保证有 n <= 10000。

整数区间 (range.cpp)

总时间限制: 1s　　内存限制: 64MB

【问题描述】

请编程完成以下任务：

1.从文件中读取闭区间的个数及它们的描述；

2.找到一个含元素个数最少的集合,使得对于每一个区间,都至少有一个整数属于该集合，输出该集合的元素个数。

【输入格式】

首行包括区间的数目 n,1<=n<=10000,接下来的 n 行,每行包括两个整数 a,b,被一空格隔开,0<=a<=b<=10000,它们是某一个区间的开始值和结束值。

【输出格式】

第一行集合元素的个数,对于每一个区间都至少有一个整数属于该区间,且集合所包含元素数目最少。

【输入样例】（range.in）

```
4
3  6
2  4
0  2
4  7
```

【输出样例】（range.out）

```
2
```