

纪念品分组(group.cpp)

总时间限制: 1s

内存限制: 64MB

【问题描述】

元旦快到了，校学生会让乐乐负责新年晚会的纪念品发放工作。为使得参加晚会的同学所获得的纪念品价值相对均衡，他要把购来的纪念品根据价格进行分组，但每组最多只能包括两件纪念品，并且每组纪念品的价格之和不能超过一个给定的整数。为了保证在尽量短的时间内发完所有纪念品，乐乐希望分组的数目最少。

你的任务是写一个程序，找出所有分组方案中分组数最少的一种，输出最少的分组数目。

【输入格式】

输入文件 group.in 包含 $n + 2$ 行：

第 1 行包括一个整数 w ，为每组纪念品价格之和的上限。

第 2 行为一个整数 n ，表示购来的纪念品的总件数。

第 3 ~ $n + 2$ 行每行包含一个正整数 p_i ($5 \leq p_i \leq w$)，表示所对应纪念品的价格。

【输出格式】

输出文件 group.out 仅一行，包含一个整数，即最少的分组数目。

【输入样例】(group.in)

```
100
9
90
20
20
30
50
60
70
80
90
```

【输出样例】(group.out)

```
6
```

【数据规模】

50%的数据满足： $1 \leq n \leq 15$ ；

100%的数据满足： $1 \leq n \leq 30000, 80 \leq w \leq 200$ 。

美元汇率(dollars.cpp)

总时间限制: 1s 内存限制: 64MB

【问题描述】

在以后的若干天里戴维将学习美元与德国马克的汇率。编写程序帮助戴维何时应买或卖马克或美元，使他从 100 美元开始，最后能获得最高可能的价值。

【输入格式】

输入文件的第一行是一个自然数 N ， $1 \leq N \leq 100$ ，表示戴维学习汇率的天数。

接下来的 N 行中每行是一个自然数 A ， $1 \leq A \leq 1000$ 。第 $i+1$ 行的 A 表示预先知道的第 $i+1$ 天的平均汇率，在这一天中，戴维既能用 100 美元买 A 马克也能用 A 马克购买 100 美元。

【输出格式】

输出文件的第一行也是唯一的一行应输出要求的钱数(单位为美元，保留两位小数)。

注意：考虑到实数算术运算中进位的误差，结果在正确结果 0.05 美元范围内的被认为是正确的，戴维必须在最后一天结束之前将他的钱都换成美元。

【输入样例】(dollars.in)

```
5
400
300
500
300
250
```

【输出样例】(dollars.out)

```
266.66
```

样例解释 (无需输出)

```
Day 1 ... changing 100.0000 美元= 400.0000 马克
Day 2 ... changing 400.0000 马克= 133.3333 美元
Day 3 ... changing 133.3333 美元= 666.6666 马克
Day 5 ... changing 666.6666 马克= 266.6666 美元
```

零件分组 (stick.cpp)

总时间限制: 1s 内存限制: 64MB

【问题描述】

某工厂生产一批棍状零件，每个零件都有一定的长度 (L_i) 和重量 (W_i)。现在为了加工需要，要将它们分成若干组，使每一组的零件都能排成一个长度和重量都不下降 (若 $i < j$ ，则 $L_i \leq L_j$ ， $W_i \leq W_j$) 的序列。请问至少要分成几组？

【输入格式】

第一行为一个整数 N ($N \leq 1000$)，表示零件的个数。第二行有 N 对正整数，每对正整数表示这些零件的长度和重量，长度和重量均不超过 10000。

【输出格式】

仅一行，即最少分成的组数。

【输入样例】(stick.in)

```
5
8 4 3 8 2 3 9 7 3 5
```

【输出样例】(stick.out)

```
2
```

运输(trans.cpp)

总时间限制: 1s

内存限制: 64MB

【问题描述】

现在已知 N 件商品。和搬运它们其中每一件的费用。现在搬家公司的老板 Mr.B 决定让我们每次任意选取 2 件商品。然后这 2 件商品只算一件商品的费用。但是这个商品的搬运费用是将选出的 2 个商品的费用之和除以 K 的运算结果。如此反复。直到只收一件商品的钱。这个就是商店要付的费用。想尽可能的少付钱，以便将更多的钱卷给希望工程。所以请你帮他计算一下最少只用付多少钱。

【输入格式】

n, k

w_1, w_2, \dots, w_n (每一件商品的搬运费用)

【输出格式】

输出一个数字，表示最少付多少钱。

【输入样例】(trans.in)

5 2

1 2 3 4 5

【输出样例】(trans.out)

1

【数据规模】

$0 < n \leq 10000$

$0 < k \leq 10000$

营养膳食(diet.cpp)

总时间限制: 1s

内存限制: 64MB

【问题描述】

阿月正在女朋友宁宁的监督下完成自己的增肥计划。

为了增肥，阿月希望吃到更多的脂肪。然而也不能只吃高脂肪食品，那样的话就会导致缺少其他营养。阿月通过研究发现：真正的营养膳食规定某类食品不宜一次性吃超过若干份。比如就一顿饭来说，肉类不宜吃超过 1 份，鱼类不宜吃超过 1 份，蛋类不宜吃超过 1 份，蔬菜类不宜吃超过 2 份。阿月想要在营养膳食的情况下吃到更多的脂肪，当然阿月的食量也是有限的。

【输入格式】

第一行包含三个正整数 n ($n \leq 200$), m ($m \leq 100$) 和 k ($k \leq 100$)。表示阿月每顿饭最多可以吃 m 份食品，同时有 n 种食品供阿月选择，而这 n 种食品分为 k 类。第二行包含 k 个不超过 10 的正整数，表示可以吃 1 到 k 类食品的最大份数。接下来 n 行每行包括 2 个正整数，分别表示该食品的脂肪指数 ai 和所属的类别 bi ，其中 $ai \leq 100$, $bi \leq k$ 。

【输出格式】

一个数字即阿月可以吃到的最大脂肪指数和。

【输入样例】(diet.in)

6 6 3

3 3 2

15 1

15 2

10 2

15 2

10 2

5 3

【输出样例】(diet.out)

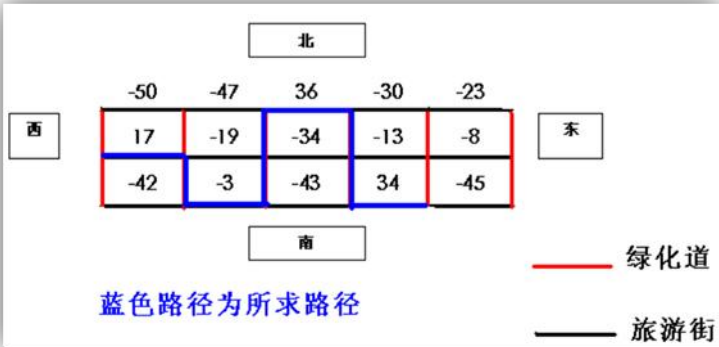
60

最佳游览线路 (travel.cpp)

总时间限制：1s 内存限制：64MB

【问题描述】

某旅游区的街道成网格状。其中东西向的街道都是旅游街，南北向的街道都是林荫道。由于游客众多，旅游街被规定为单行道，游客在旅游街上只能从西向东走，在林荫道上则既可从南向北走，也可以从北向南走。阿龙想到这个旅游区游玩。他的好友阿福给了他一些建议，用分值表示所有旅游街相邻两个路口之间的街道值得游览的程度，分值时从-100 到 100 的整数，所有林荫道不打分。所有分值不可能全是负分。例如下图是被打过分的某旅游区的街道图：



阿龙可以从任一个路口开始游览，在任一个路口结束游览。请你写一个程序，帮助阿龙找一条最佳的游览线路，使得这条线路的所有分值总和最大。

【输入格式】

第一行是两个整数 M 和 N ，之间用一个空格符隔开， M 表示有多少条旅游街 ($1 \leq M \leq 100$)， N 表示有多少条林荫道 ($1 \leq N \leq 20001$)。接下来的 M 行依次给出了由北向南每条旅游街的分值信息。每行有 $N-1$ 个整数，依次表示了自西向东旅游街每一小段的分值。同一行相邻两个数之间用一个空格隔开。

【输出格式】

只有一行，是一个整数，表示你的程序找到的最佳游览线路的总分值。

【输入样例】(travel.in)

```
3 6
-50 -47 36 -30 -23
17 -19 -34 -13 -8
-42 -3 -43 34 -45
```

【输出样例】(travel.out)

84

一个简单的问题 (easy.cpp)

总时间限制: 1s

内存限制: 64MB

【问题描述】

给定一个正整数 n ，您的任务是找出一个正整数 m ，它是大于 n 的最小整数，并且 m 的二进制形式与 n 的二进制形式有相同数目的 1。例如，如果给出“78”，我们可以写出它的二进制形式，“1001110”。这个二进制形式有 4 个 1。大于“1001110”并且也包含 4 个 1 的最小整数是“1010011”，即“83”，因此应该输出“83”。

【输入格式】

若干行，每行一个整数，即 n ($1 \leq n \leq 1000000$)。输入“0”则结束。

【输出格式】

每行一个整数，即 m 。

【输入样例】(easy.in)

```
1
2
3
4
78
0
```

【输出样例】(easy.out)

```
2
4
5
8
83
```

金银岛 (island.cpp)

总时间限制: 1s

内存限制: 64MB

【问题描述】

某天 KID 利用飞行器飞到了一个金银岛上，上面有许多珍贵的金属，KID 虽然更喜欢各种宝石的艺术品，可是也不拒绝这样珍贵的金属。但是他只带着一个口袋，口袋至多只能装重量为 w 的物品。岛上金属有 s 个种类，每种金属重量不同，分别为 n_1, n_2, \dots, n_s ，同时每个种类的金属总的价值也不同，分别为 v_1, v_2, \dots, v_s 。KID 想一次带走价值尽可能多的金属，问他最多能带走价值多少的金属。注意到金属是可以被任意分割的，并且金属的价值和其重量成正比。

【输入格式】

第 1 行是测试数据的组数 k ，后面跟着 k 组输入。

每组测试数据占 3 行，第 1 行是一个正整数 w ($1 \leq w \leq 10000$)，表示口袋承重上限。第 2 行是一个正整数 s ($1 \leq s \leq 100$)，表示金属种类。第 3 行有 $2s$ 个正整数，分别为 $n_1, v_1, n_2, v_2, \dots, n_s, v_s$ 分别为第一种，第二种，...，第 s 种金属的总重量和总价值 ($1 \leq n_i \leq 10000, 1 \leq v_i \leq 10000$)。

【输出格式】

k 行，每行输出对应一个输入。输出应精确到小数点后 2 位。

【输入样例】(island.in)

```
2
50
4
10 100 50 30 7 34 87 100
10000
5
1 43 43 323 35 45 43 54 87 43
```

【输出样例】(island.out)

```
171.93
508.00
```