

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	金币	扫雷游戏	求和	推销员
英文题目与子目录名	coin	mine	sum	salesman
可执行文件名	coin	mine	sum	salesman
输入文件名	coin.in	mine.in	sum.in	salesman.in
输出文件名	coin.out	mine.out	sum.out	salesman.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
运行内存上限	128M	128M	128M	128M

二. 提交源程序文件名

对于 C++ 语言	coin.cpp	mine.cpp	sum.cpp	salesman.cpp
对于 C 语言	coin.c	mine.c	sum.c	salesman.c
对于 pascal 语言	coin.pas	mine.pas	sum.pas	salesman.pas

三. 编译命令 (不包含任何优化开关)

对于 C++ 语言	g++ -o coin coin.cpp -lm	g++ -o mine mine.cpp -lm	g++ -o sum sum.cpp -lm	g++ -o salesman salesman.cpp -lm
对于 C 语言	gcc -o coin coin.c -lm	gcc -o mine mine.c -lm	gcc -o sum sum.c -lm	gcc -o salesman salesman.c -lm
对于 pascal 语言	fpc coin.pas	fpc mine.pas	fpc sum.pas	fpc salesman.pas

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 AMD Athlon(tm) II x2 240 processor, 2.8GHz, 内存 4G, 上述时限以此配置为准。
- 4、只提供 Linux 格式附加样例文件。
- 5、特别提醒: 评测在当前最新公布的 NOI Linux 下进行, 各语言的编译器版本以其为准。

1. 金币

(coin.cpp/c/pas)

【问题描述】

国王将金币作为工资，发放给忠诚的骑士。第一天，骑士收到一枚金币；之后两天（第二天和第三天），每天收到两枚金币；之后三天（第四、五、六天），每天收到三枚金币；之后四天（第七、八、九、十天），每天收到四枚金币……；这种工资发放模式会一直这样延续下去：当连续 N 天每天收到 N 枚金币后，骑士会在之后的连续 $N+1$ 天里，每天收到 $N+1$ 枚金币。

请计算在前 K 天里，骑士一共获得了多少金币。

【输入格式】

输入文件名为 coin.in。

输入文件只有 1 行，包含一个正整数 K ，表示发放金币的天数。

【输出格式】

输出文件名为 coin.out。

输出文件只有 1 行，包含一个正整数，即骑士收到的金币数。

【输入输出样例 1】

coin.in	coin.out
6	14

见选手目录下的 coin/coin1.in 和 coin/coin1.ans。

【输入输出样例 1 说明】

骑士第一天收到一枚金币；第二天和第三天，每天收到两枚金币；第四、五、六天，每天收到三枚金币。因此一共收到 $1+2+2+3+3+3=14$ 枚金币。

【输入输出样例 2】

coin.in	coin.out
1000	29820

见选手目录下的 coin/coin2.in 和 coin/coin2.ans。

【数据说明】

对于 100% 的数据， $1 \leq K \leq 10,000$ 。

2. 扫雷游戏

(mine.cpp/c/pas)

【问题描述】

扫雷游戏是一款十分经典的单机小游戏。在 n 行 m 列的雷区中有一些格子含有地雷（称之为地雷格），其他格子不含地雷（称之为非地雷格）。玩家翻开一个非地雷格时，该格将会出现一个数字——提示周围格子中有多少个是地雷格。游戏的目标是在不翻出任何地雷格的条件下，找出所有的非地雷格。

现在给出 n 行 m 列的雷区中的地雷分布，要求计算出每个非地雷格周围的地雷格数。

注：一个格子的周围格子包括其上、下、左、右、左上、右上、左下、右下八个方向上与之直接相邻的格子。

【输入格式】

输入文件名为 mine.in。

输入文件第一行是用一个空格隔开的两个整数 n 和 m ，分别表示雷区的行数和列数。

接下来 n 行，每行 m 个字符，描述了雷区中的地雷分布情况。字符 '*' 表示相应格子是地雷格，字符 '?' 表示相应格子是非地雷格。相邻字符之间无分隔符。

【输出格式】

输出文件名为 mine.out。

输出文件包含 n 行，每行 m 个字符，描述整个雷区。用 '*' 表示地雷格，用周围的地雷个数表示非地雷格。相邻字符之间无分隔符。

【输入输出样例 1】

mine.in	mine.out
3 3 *?? ??? ?*?	*10 221 1*1

见选手目录下的 mine/mine1.in 和 mine/mine1.ans。

【输入输出样例 2】

mine.in	mine.out
2 3 ?*? *??	2*1 *21

见选手目录下的 mine/mine2.in 和 mine/mine2.ans。

【输入输出样例 3】

见选手目录下的 mine/mine3.in 和 mine/mine3.ans。

【数据说明】

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

3. 求和

(sum.cpp/c/pas)

【问题描述】

一条狭长的纸带被均匀划分出了 n 个格子，格子编号从 1 到 n 。每个格子上都染了一种颜色 $color_i$ （用 $[1, m]$ 中的一个整数表示），并且写了一个数字 $number_i$ 。

	5	5	3	2	2	2
编号	1	2	3	4	5	6

定义一种特殊的三元组：(x, y, z)，其中 x, y, z 都代表纸带上格子的编号，这里的三元组要求满足以下两个条件：

1. x, y, z 都是整数, $x < y < z$, $y - x = z - y$
2. $color_x = color_z$

满足上述条件的三元组的分数规定为 $(x + z) * (number_x + number_z)$ 。整个纸带的分数规定为所有满足条件的三元组的分数的和。这个分数可能会很大，你只要输出整个纸带的分数除以 10,007 所得的余数即可。

【输入格式】

输入文件名为 sum.in。

第一行是用一个空格隔开的两个正整数 n 和 m ， n 代表纸带上格子的个数， m 代表纸带上颜色的种类数。

第二行有 n 个用空格隔开的正整数，第 i 个数字 $number_i$ 代表纸带上编号为 i 的格子上面写的数字。

第三行有 n 个用空格隔开的正整数，第 i 个数字 $color_i$ 代表纸带上编号为 i 的格子染的颜色。

【输出格式】

输出文件名为 sum.out。

共一行，一个整数，表示所求的纸带分数除以 10,007 所得的余数。

【输入输出样例 1】

sum.in	sum.out
6 2 5 5 3 2 2 2 2 2 1 1 2 1	82

见选手目录下的 sum/sum1.in 和 sum/sum1.ans。

【输入输出样例 1 说明】

纸带如题目描述中的图所示。

所有满足条件的三元组为：(1, 3, 5), (4, 5, 6)。

所以纸带的分数为 $(1 + 5) * (5 + 2) + (4 + 6) * (2 + 2) = 42 + 40 = 82$ 。

【输入输出样例 2】

sum.in	sum.out
15 4 5 10 8 2 2 2 9 9 7 7 5 6 4 2 4 2 2 3 3 4 3 3 2 4 4 4 4 1 1 1	1388

见选手目录下的 sum/sum2.in 和 sum/sum2.ans。

【输入输出样例 3】

见选手目录下的 sum/sum3.in 和 sum/sum3.ans。

【数据说明】

对于第 1 组至第 2 组数据， $1 \leq n \leq 100, 1 \leq m \leq 5$ ；

对于第 3 组至第 4 组数据， $1 \leq n \leq 3000, 1 \leq m \leq 100$ ；

对于第 5 组至第 6 组数据， $1 \leq n \leq 100000, 1 \leq m \leq 100000$ ，且不存在出现次数超过 20 的颜色；

对于全部 10 组数据， $1 \leq n \leq 100000, 1 \leq m \leq 100000, 1 \leq color_i \leq m, 1 \leq number_i \leq 100000$ 。

4. 推销员

(salesman.cpp/c/pas)

【问题描述】

阿明是一名推销员，他奉命到螺丝街推销他们公司的产品。螺丝街是一条死胡同，出口与入口是同一个，街道的一侧是围墙，另一侧是住户。螺丝街一共有 N 家住户，第 i 家住户到入口的距离为 S_i 米。由于同一栋房子里可以有多家住户，所以可能有多家住户与入口的距离相等。阿明会从入口进入，依次向螺丝街的 X 家住户推销产品，然后再原路走出去。

阿明每走 1 米就会积累 1 点疲劳值，向第 i 家住户推销产品会积累 A_i 点疲劳值。阿明是工作狂，他想知道，对于不同的 X ，在不走多余的路的前提下，他最多可以积累多少点疲劳值。

【输入格式】

输入文件名为 salesman.in。

第一行有一个正整数 N ，表示螺丝街住户的数量。

接下来的一行有 N 个正整数，其中第 i 个整数 S_i 表示第 i 家住户到入口的距离。数据保证 $S_1 \leq S_2 \leq \dots \leq S_n < 10^8$ 。

接下来的一行有 N 个正整数，其中第 i 个整数 A_i 表示向第 i 户住户推销产品会积累的疲劳值。数据保证 $A_i < 10^3$ 。

【输出格式】

输出文件名为 salesman.out。

输出 N 行，每行一个正整数，第 i 行整数表示当 $X=i$ 时，阿明最多积累的疲劳值。

【输入输出样例 1】

salesman.in	salesman.out
5	15
1 2 3 4 5	19
1 2 3 4 5	22
	24
	25

见选手目录下的 salesman/salesman1.in 和 salesman/salesman1.ans。

【输入输出样例 1 说明】

$X=1$: 向住户 5 推销，往返走路的疲劳值为 $5+5$ ，推销的疲劳值为 5，总疲劳值为 15。

$X=2$: 向住户 4、5 推销，往返走路的疲劳值为 $5+5$ ，推销的疲劳值为 $4+5$ ，总疲劳值为 $5+5+4+5=19$ 。

$X=3$: 向住户 3、4、5 推销，往返走路的疲劳值为 $5+5$ ，推销的疲劳值 $3+4+5$ ，总疲劳值为 $5+5+3+4+5=22$ 。

$X=4$: 向住户 2、3、4、5 推销，往返走路的疲劳值为 $5+5$ ，推销的疲劳值 $2+3+4+5$ ，总疲劳值 $5+5+2+3+4+5=24$ 。

$X=5$: 向住户 1、2、3、4、5 推销，往返走路的疲劳值为 $5+5$ ，推销的疲劳值 $1+2+3+4+5$ ，总疲劳值 $5+5+1+2+3+4+5=25$ 。

【输入输出样例 2】

salesman.in	salesman.out
5	12
1 2 2 4 5	17
5 4 3 4 1	21
	24
	27

见选手目录下的 salesman/salesman2.in 和 salesman/salesman2.ans。

【输入输出样例 2 说明】

X=1: 向住户 4 推销, 往返走路的疲劳值为 $4+4$, 推销的疲劳值为 4, 总疲劳值 $4+4+4=12$ 。

X=2: 向住户 1、4 推销, 往返走路的疲劳值为 $4+4$, 推销的疲劳值为 $5+4$, 总疲劳值 $4+4+5+4=17$ 。

X=3: 向住户 1、2、4 推销, 往返走路的疲劳值为 $4+4$, 推销的疲劳值为 $5+4+4$, 总疲劳值 $4+4+5+4+4=21$ 。

X=4: 向住户 1、2、3、4 推销, 往返走路的疲劳值为 $4+4$, 推销的疲劳值为 $5+4+3+4$, 总疲劳值 $4+4+5+4+3+4=24$ 。或者向住户 1、2、4、5 推销, 往返走路的疲劳值为 $5+5$, 推销的疲劳值为 $5+4+4+1$, 总疲劳值 $5+5+5+4+4+1=24$ 。

X=5: 向住户 1、2、3、4、5 推销, 往返走路的疲劳值为 $5+5$, 推销的疲劳值为 $5+4+3+4+1$, 总疲劳值 $5+5+5+4+3+4+1=27$ 。

【样例输入输出 3】

见选手目录下的 salesman/salesman3.in 和 salesman/salesman3.ans。

【数据说明】

对于 20% 的数据, $1 \leq N \leq 20$;

对于 40% 的数据, $1 \leq N \leq 100$;

对于 60% 的数据, $1 \leq N \leq 1000$;

对于 100% 的数据, $1 \leq N \leq 100000$ 。