

CSP-X2022 小学组二轮补赛试题（上半场）

考试时间：2023 年 3 月 4 日下午 14: 30--16: 00

题目名称	独木桥	移动棋子
题目类型	传统型	传统型
目录	bridge	chess
程序名称	bridge.cpp	chess.cpp
输入文件名	bridge.in	chess.in
输出文件名	bridge.out	chess.out
测试点数量	10	10
每测试点时限	1 秒	1 秒
每测试点分值	10	10
内存限制	512M	512M

注意事项

1、代码必须放在子文件夹内，子文件夹名与题目英文名一致。文件名（包括程序名和输入输出文件名）必须使用英文小写。

2、C++编译选项：-O2 -std=c++14。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、若无特殊说明，输入文件中同一行内的多个整数、浮点数、字符串等均使用一个空格分隔。若无特殊说明，结果比较方式为忽略行末空格、文末回车后的全文比较。

第一题 独木桥 (bridge)

题目描述

长度为 L 米的独木桥上有 n 个人，他们每个人都想以最快的时间离开危险的独木桥。

已知每个人在独木桥上的行走速度为 1 米/秒，每个人只要能走到独木桥的两个端点中的其中一个就可以离开独木桥。

由于独木桥的桥面宽度很窄，只能容纳一个人通过，当两个人相遇时，他们无法交错通过，只能各自调转方向，继续沿反方向行走。

给你独木桥上的人数 n ，独木桥的长度 L ，第 i 个人的初始位置到独木桥左端点的距离 a_i 米（每个人开始的朝向未知，但他们可以根据需要随时调转行走的方向）。

请计算出所有人同时出发，全部都离开独木桥所需的最短时间。

输入格式（输入文件为 bridge.in）

第一行一个整数 n ，表示人数。

第二行一个整数 L ，表示独木桥的长度（米）。

第三行是 $a_1, a_2, \dots, a_n$ ，其中 a_i 表示第 i 个人初始位置到独木桥左端点的距离。

输出格式（输出文件为 bridge.out）

输出一行一个整数，表示所有人都离开独木桥所需的最短时间。

输入输出样例

样例 1 输入

```
3
10
2 6 7
```

样例 1 输出

```
4
```

说明：三个人同时出发，第一个人向左走，需要 2 秒离开桥，第二个人向右走需要 4 秒离开桥，第三个人向右走需要 3 秒离开桥。所以，4 秒后，三个人都离开了独木桥。

样例 2 输入

```
7
214
11 12 7 13 176 23 191
```

样例 2 输出

```
38
```

数据范围:

- 对于 50% 的数据: $1 \leq n \leq 10^3$;
- 对于 100% 的数据: $1 \leq n \leq 10^6$, $1 \leq L \leq 10^6$, $0 \leq a_i \leq L$ 。

第二题 移动棋子 (chess)

题目描述

一维的棋盘上有无限多个格子, 每个格子都有一个编号, 最中间的格子编号为 0, 0 号格子向右依次编号为 1, 2, 3, ..., 向左依次编号为 -1, -2, -3, ...。

小明的目标是要将一枚棋子从 x 号格子移动到 y 号格子。

每一次操作有两种选择:

操作 1: 向右移动 1 个格子。

操作 2: 从当前棋子所在的 a 号格子, 直接跳到 $-a$ 号格子 (如: 可以从 6 直接跳到 -6, 也可以从 -6 直接跳到 6)。

可以证明, 无论整数 x 和 y 的值是多少, 目标总是可以实现的。

请你设计程序, 帮小明计算把棋子从 x 号格子移动到 y 号格子需要的最少操作次数。

输入格式 (输入文件为 chess.in)

一行, 两个整数 x 和 y , 表示要将棋子从 x 号格子移动到 y 号格子。

输出格式 (输出文件为 chess.out)

一个整数, 表示小明把棋子从 x 号格子移动到 y 号格子需要的最少操作次数。

输入输出样例

样例 1 输入

10 20

样例 1 输出

10

样例 1 说明：需要进行 10 次操作 1。

样例 2 输入

10 -10

样例 2 输出

1

样例 2 说明：需要进行 1 次操作 2。

样例 3 输入

-10 -20

样例 3 输出

12

样例 3 说明：先进行 1 次操作 2，然后进行 10 次操作 1，最后进行 1 次操作 2。

数据范围

- 30% 的数据： $|x| \leq 100, |y| \leq 100$ 。
- 100% 的数据： x, y 都为整数； $|x| \leq 10^9, |y| \leq 10^9$ ； x, y 不相等。
- 有 40% 的数据， $|x| > |y|$ 。
- 有 40% 的数据， $|x| < |y|$ 。
- 有 20% 的数据， $|x| = |y|$ 。

CSP-X2022 小学组二轮补赛试题（下半场）

考试时间：2023 年 3 月 4 日下午 16: 30--18: 00

题目名称	动物园	摧毁
题目类型	传统型	传统型
目录	zoo	destroy
程序名称	zoo.cpp	destroy.cpp
输入文件名	zoo.in	destroy.in
输出文件名	zoo.out	destroy.out
测试点数量	10	10
每测试点时限	1 秒	2 秒
每测试点分值	10	10
内存限制	512M	512M

注意事项

1、代码必须放在子文件夹内，子文件夹名与题目英文名一致。文件名（包括程序名和输入输出文件名）必须使用英文小写。

2、C++编译选项：-O2 -std=c++14。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、若无特殊说明，输入文件中同一行内的多个整数、浮点数、字符串等均使用一个空格分隔。若无特殊说明，结果比较方式为忽略行末空格、文末回车后的全文比较。

第一题 动物园 (zoo)

题目描述

某动物园里有 n 个场馆和 m 种动物 ($m \leq n$)。

n 个场馆的编号分别用 $1, 2, 3, \dots, n$ 表示; m 种动物的编号分别用 $1, 2, 3, \dots, m$ 表示。每一个场馆中只饲养了一只动物, 不同的场馆可能饲养着相同种类的动物。

这个动物园的门票比较特殊, 游客在购买门票时必须说明要参观的场馆的起止编号 a 和 b (起止编号会打印到游客购买的门票上), 代表游客只能参观动物园的第 a 个场馆至第 b 个场馆 (包含 a, b) 里的动物, 其他的场馆不能去。门票按一个场馆十元收费。

如果你购买的门票的起止场馆编号是 3 到 8, 那么你需要花 60 元钱购买门票, 只能观看 3, 4, 5, 6, 7, 8 号场馆的动物。

小明希望看到动物园内所有种类的动物, 同时小明是个非常节约的孩子, 他希望花最少的钱买门票。请你帮小明计算: 他最少需要花费多少钱买门票才能看到所有种类的动物 (同一种动物他可能不止看一个)。注意: 小明只能买一张门票。

输入格式 (输入文件为 zoo.in)

第一行两个整数 n, m , 分别表示动物园内的场馆数量及动物种类数量。

第二行是 $x_1, x_2, \dots, x_n$, 其中 x_i 表示第 i 个场馆中的动物种类编号。

输出格式 (输出文件为 zoo.out)

一行一个整数 p , 表示小明的门票费用。

输入输出样例

样例 1 输入

```
12 5
2 5 3 1 3 2 4 1 1 5 4 3
```

样例 1 输出

```
60
```

样例 1 说明: 花费最少的其中一种购票方案选择是 $a = 2, b = 7$, 表示购买场馆 2, 3, 4, 5, 6, 7 的门票, 分别看到的动物是 5, 3, 1, 3, 2, 4, 其中动物 3 小明看了两个。

数据范围

- 对于 30% 的数据, 有 $n \leq 200, m \leq 20$ 。
- 对于 60% 的数据, 有 $n \leq 1000, m \leq 1000$ 。
- 对于 100% 的数据, 有 $1 \leq n \leq 10^6, 1 \leq x_i \leq m \leq 2 \times 10^3$ 。

第二题 摧毁 (destroy)

题目描述

坐地日行八万里，巡天遥看一千河。

2077 年，人类不仅仅是赛博科技得到了发展，太空技术也已经得到了极大的发展。地球的不同外轨道上已经充斥着各种功能用途的人造卫星。因为一个轨道上的卫星数量是有上限的，且卫星更新换代速度很快，如果想要发射新的卫星，需要把所有旧的卫星摧毁。

人类有两种不同的武器可以摧毁卫星，具体如下（其中 PW 为新的能量单位）：

- (1) 使用定点激光武器花费 $1\ PW$ 的代价摧毁任意轨道上指定的一个卫星。
- (2) 使用脉冲轨道武器花费 $c\ PW$ 的代价把某一轨道上的所有卫星摧毁。

现在有 n 个旧卫星分布在不同的外轨道上，你的任务是摧毁这些旧卫星。给出这 n 个卫星的轨道编号，求将这些卫星全部摧毁的最小代价是多少？

输入格式（输入文件名为 `destroy.in`）

第一行一个正整数 T ，表示测试数据组数。

接下来对于每组测试数据（注意：每组测试数据有 2 行数据，以下共 $2 \cdot T$ 行数据）：

第一行两个正整数 n 和 c 表示需要摧毁的卫星数量和使用脉冲轨道武器的代价。

第二行是 $x_1, x_2, \dots, x_n$ ，其中 x_i 表示第 i 个卫星的轨道编号。

输出格式（输出文件名为 `destroy.out`）

输出 T 行答案，对于每组测试数据，输出一行一个整数表示摧毁所有卫星的代价。

输入输出样例

样例 1 输入

```
4
10 1
2 1 4 5 2 4 5 5 1 2
5 2
3 2 1 2 2
2 2
1 1
2 2
1 2
```

样例 1 输出

```
4
4
```

2
2

样例说明：对于第一组测试数据，使用脉冲武器的代价为 1 PW。轨道 1 上有 2 个卫星，轨道 2 上有 3 个卫星，轨道 4 上有 2 个卫星，轨道 5 上有 3 个卫星。因此对于轨道 1、2、4、5，均使用脉冲武器各花费 1PW 的代价可全部摧毁，总的代价为 4 PW，很显然该方案为总代价最小方案。

对于第二组测试数据，使用脉冲武器的代价为 2 PW。轨道 1 上有 1 个卫星，轨道 2 上有 3 个卫星，轨道 3 上有 1 个卫星。因此，对于轨道 1 采用激光武器，轨道 2 采用脉冲武器，轨道 3 采用激光武器可全部摧毁所有卫星，总的代价为 4 PW，很显然该方案使得总代价最小。

数据范围

对于 30% 的数据， $T = 1, 1 \leq n \leq 10, 1 \leq a_i \leq 10, 1 \leq c \leq 10$;

对于 60% 的数据， $1 \leq n \leq 10^3, 1 \leq a_i \leq 1000, 1 \leq c \leq 100$;

对于 100% 的数据， $1 \leq T \leq 10, 1 \leq n \leq 10^6, 1 \leq a_i \leq 10^6, 1 \leq c \leq 100$ ，且所有测试数据的 n 加起来不超过 10^6 。