CSP-X2023 山东小学组第二轮试题（下半场）

考试时间： 2023 年 10 月 22 日上午 10：30--12：00

题目名称	克隆机	代价
题目类型	传统型	传统型
子文件夹名	clone	cost
程序名称	clone.cpp	cost.cpp
输入文件名	clone.in	cost.in
输出文件名	clone.out	cost.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。
4、选手提交的程序源文件不能大于 100KB。
5、程序使用的栈空间内存限制与题目的内存限制要求一致。

克隆机（clone）

【题目描述】

有一台神奇的克隆机，可以克隆任何东西。将样品放进克隆机，可以克隆出一份一样的“复制品”。小明得到了 k 种珍贵的植物种子，依次用 A,B,C,D,...,Z 表示（1≤k≤26）。一开始，每种植物种子只有 1 粒。

小明想利用克隆机克隆出更多种子。将一粒种子作为样品放进克隆机，就可以得到一粒克隆出来的相同的种子，这样一粒种子就变成了两粒种子。小明将 k 粒不同的种子按字母先后顺序排队，从 A 开始依次放入克隆机，每次把得到的两粒相同的种子（放入的 1 粒和克隆出来的 1 粒）放到队尾，这样不断的进行克隆。

例如，一共有 7 种不同的种子，依次用 A,B,C,D,E,F,G 表示。

第 1 粒种子 A 放进克隆机之前，队列是：A,B,C,D,E,F,G。

第 1 粒种子 A 放进克隆机之后，队列是：B,C,D,E,F,G,A,A。

第 3 粒种子 C 放进克隆机之前，队列是：C,D,E,F,G,A,A,B,B 。

第 3 粒种子 C 放进克隆机之后，队列是：D,E,F,G,A,A,B,B,C,C 。

请问第 n 粒放进克隆机的是什么种子？用 A,B,...,Z 表示。

【输入格式】

输入文件名为 clone.in。

输入 1 行 2 个数字，k 和 n ，用空格隔开。

【输出格式】

输出文件名为 clone.out。

输出 1 个字符，代表第 n 粒放进克隆机的种子。

【样例 1 输入】

```
7 10
```

【样例 1 输出】

```
B
```

【样例 1 解释】

序号	1	2	3	4	5	6	7	8	9	10	11
种子	A	B	C	D	E	F	G	A	A	B	B

【样例 2 输入】

```
26 80
```

【样例 2 输出】

```
A
```

【样例 3 输入】

```
15 689
```

【样例 3 输出】

```
G
```

【数据范围】

对于 50% 的数据，$1 \leqslant n \leqslant 10^6$ ；

对于 100% 的数据，$1 \leqslant k \leqslant 26$，$1 \leqslant n \leqslant 10^{18}$ 。

代价（cost）

【题目描述】

因为“黑发不知勤学早”，于是小明成为了一名伟大的流水线工人，天天起早摸黑打螺丝。

这一天，小明所在的流水线生成了 n 件产品，其中第 i 号产品规格用一个正整数 a_i 表示。

所谓流水线，就是需要标准化。于是，小明想把这 n 件产品规格修整得全部相同。

小明手边有两种工具来进对产品进行修整，但是使用不同工具需要花费不同的代价，小明可以进行以下操作任意次：

- 使用一次第一种工具花费 A 的代价将第 i 件产品的规格 a_i 修改成 a_i+1（其中 $i\in[1,n]$）。
- 使用一次第二种工具花费 B 的代价将第 i 件产品的规格 a_i 修改成 a_i-1（其中 $i\in[1,n]$）。

现在小明想要花费最少的代价将所有产品的规格都变得相同，于是他找到了自幼勤学苦练的你来帮忙。

你只需要计算出把所有产品调整为相同规格的最小代价即可。

【输入格式】

输入文件为 cost.in。

第一行三个正整数 n,A,B，分别表示产品数量，使用一次第一种工具的代价 A 和使用一次第二种工具的代价 B。

第二行 n 个正整数 $a_1,a_2,\ldots,a_n$ 表示每件产品的产品规格。

【输出格式】

输出文件为 cost.out。

一行一个整数表示最小的总代价。

【样例 1 输入】

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

【样例 1 输出】

```
4
```

【样例 1 解释】

两种操作的代价相等，所以把所有产品规格修改成 2 花费的代价最小，计算可得最小代价为 4（1 变为 2，5 变为 4，4 再变为 3，3 再变为 2，已经规格相同，共 4 次）。

【样例 2 输入】

```
3 1 100
1 2 5
```

【样例 2 输出】

```
7
```

【样例 2 解释】

因为二操作代价 B 太大，所以把所有产品规格修改成 5 花费代价最小，计算可得最小代价为 7（用一操作，1 变为 5 需要 4 次，2 变为 5 需要 3 次，共 7 次）。

【样例 3 输入】

```
3 2 5
9999999999 9999999999 9999999999
```

【样例 3 输出】

```
0
```

【数据范围】

对于 30% 的数据，$1 \le n \le 10, 1 \le a_i \le 100, 1 \le A, B \le 10$；

对于 60% 的数据，$1 \le n \le 10^5, 1 \le a_i \le 10^5, 1 \le A, B \le 100$；

其中有 30% 的数据，$A = B$；

对于 100% 的数据，$1 \le n \le 10^5, 0 \le a_i \le 10^9, 1 \le A, B \le 1000$。