

CSP-J 国庆强化营 · 全真模拟赛

Day 4

2026 年 10 月 4 日上午 9:00-12:00

题目名称	程老师的借阅登记机	程老师的工牌校验	程老师的周转交接	程老师的跳格训练
文件名	borrow.cpp	badge.cpp	stock.cpp	jump.cpp
输入文件	borrow.in	badge.in	stock.in	jump.in
输出文件	borrow.out	badge.out	stock.out	jump.out
时间限制	1000ms	1000ms	1000ms	1000ms
空间限制	512MB	512MB	512MB	512MB

注意事项

- 每道题提交一个源文件，文件名见上表（全小写，如 borrow.cpp）。
- 程序一律使用文件读写：从 *.in 读入，向 *.out 输出，不使用标准输入输出。
- 编译命令：g++ -O2 -std=c++14；评测环境为 NoiLinux 2.0。
- 时间限制与空间限制见上表，程序运行耗时与内存以评测机为准。

程老师的借阅登记机

文件读写

- 输入文件 borrow.in
- 输出文件 borrow.out

限制

- 1000ms
- 512MB

题目描述

程老师把班里的图书角交给一台登记机管理。老师每借出一本书或收回一本书，都会在登记机上留下一条记录：借书记作 B_x ，还书记作 R_x ，其中 x 是书号。登记机依次处理这些记录，处理规则只有两条：

- 处理借书记录 B_x 时，若书 x 当前已经处于借出状态（借出后尚未归还），这条记录会被**拒绝**，不计入成功次数；否则借书成功，书 x 转为借出状态。
- 处理还书记录 R_x 时，若书 x 当前不处于借出状态，这条记录会被**忽略**，不计入成功次数；否则还书成功，书 x 转回在架状态。

所有记录处理完毕后，老师想知道两件事：一共有多少条记录被成功执行；以及此时仍然借出在外的是哪些书。

输入格式

从文件 borrow.in 中读入数据。

第一行一个正整数 n ，表示记录的条数。

接下来 n 行，每行一个字符和一个整数，中间用一个空格隔开。字符为 B 或 R，表示这是一次借书或还书操作；整数 x 表示书号。

输出格式

输出到文件 borrow.out 中。

第一行一个整数，表示被成功执行的记录条数。

第二行输出此时仍处于借出状态的书号，按**从小到大**的顺序排列，相邻两个书号之间用一个空格隔开。若没有任何书处于借出状态，第二行为空行。

数据范围

对于所有测试数据，保证： $1 \leq n \leq 10^5$ ， $1 \leq x \leq 10^4$ 。

测试点	$n \leq$	特殊性质
1~4	10	无
5~10	10^3	无
11~16	10^5	无
17~18	10^5	无
19	10^5	A
20	10^5	B

特殊性质 A：输入中不存在被拒绝或被忽略的记录，每条记录都执行成功。

特殊性质 B：所有记录的书号都不超过 10。

样例输入 1

```
6
B 1001
R 1001
B 1001
B 1001
R 1002
B 1003
```

样例输出 1

```
4
1001 1003
```

样例输入 2

```
3
B 7
B 7
R 8
```

样例输出 2

```
1
7
```

样例解释

样例 1：第 1 条 B 1001 执行成功，1001 借出；第 2 条 R 1001 执行成功，1001 归还；第 3 条 B 1001 再次借出成功；第 4 条 B 1001 因 1001 已在借出状态被拒绝；第 5 条 R 1002 因 1002 不在借出状态

被忽略；第 6 条 B 1003 执行成功。成功记录共 4 条，最终 1001、1003 借出未还，按升序输出。

样例 2：第 1 条 B 7 成功；第 2 条 B 7 因 7 已借出被拒；第 3 条 R 8 因 8 未借出被忽略。仅 1 条成功，最终借出的是 7。

程老师的工牌校验

文件读写

- 输入文件 badge.in
- 输出文件 badge.out

限制

- 1000ms
- 512MB

题目描述

学校为每位老师制作工牌。工牌号由 **6 位数字本体** 和紧随其后的 **1 位校验符** 组成，共 7 位。校验符按如下规则计算：设本体从左到右的 6 个数字依次为 $d_1, d_2, \dots, d_6$ ，先计算

$$v = (2d_1 + 3d_2 + 5d_3 + 7d_4 + 11d_5 + 13d_6) \bmod 13$$

即把 6 个数字分别乘以 2, 3, 5, 7, 11, 13 后求和，再对 13 取余。校验符按 v 的值书写： v 为 $0 \sim 9$ 时直接写作对应的数字字符； v 为 10 时写作 A， v 为 11 时写作 B， v 为 12 时写作 C。

人事部门给了老师一份待录入的本体清单，每个本体是恰好 6 个数字组成的字符串，**可能含前导零**（前导零既参与校验计算，也在工牌号中原样保留）。请为清单中的每个本体输出完整的工牌号。

输入格式

从文件 badge.in 中读入数据。

第一行一个正整数 q ，表示本体个数。

接下来 q 行，每行一个长度为 6 的数字字符串，表示一个本体。

输出格式

输出到文件 badge.out 中。

q 行，每行一个工牌号，即本体 6 个字符后接上该校验符。

数据范围

对于所有测试数据，保证： $1 \leq q \leq 10^5$ ，每个本体都是长度恰好为 6 的数字字符串。

测试点	$q \leq$	特殊性质
1~4	10	无
5~8	10^3	无
9~14	10^5	无

测试点	$q \leq$	特殊性质
15~18	10^5	无
19	10^5	A
20	10^5	B

特殊性质 A：所有本体的校验符都是 A、B 或 C（即每个 v 都不小于 10）。

特殊性质 B：所有本体的首位都是 0（即每个本体都含前导零）。

样例输入 1

```
3
001234
000010
135791
```

样例输出 1

```
0012340
000010B
1357912
```

样例输入 2

```
2
999999
100000
```

样例输出 2

```
9999995
1000002
```

样例解释

样例 1：本体 001234 的加权和为 $2 \times 0 + 3 \times 0 + 5 \times 1 + 7 \times 2 + 11 \times 3 + 13 \times 4 = 104$, $104 \bmod 13 = 0$, 校验符为 0, 工牌号 0012340; 本体 000010 的加权和为 11, $11 \bmod 13 = 11$, 校验符 B, 工牌号 000010B; 本体 135791 的加权和为 $2 + 9 + 25 + 49 + 99 + 13 = 197$, $197 \bmod 13 = 2$, 校验符 2, 工牌号 1357912。

样例 2：本体 999999 的加权和为 $9 \times (2 + 3 + 5 + 7 + 11 + 13) = 369$, $369 \bmod 13 = 5$, 工牌号 9999995; 本体 100000 只有第 1 位非零, 加权和为 $2 \times 1 = 2$, $2 \bmod 13 = 2$, 工牌号 1000002。

程老师的周转交接

文件读写

- 输入文件 stock.in
- 输出文件 stock.out

限制

- 1000ms
- 512MB

题目描述

教具仓库与 n 个班组有固定的交接顺序：第 1 个班组先来，第 2 个班组随后，一直到第 n 个班组。每个班组来时与仓库做一次交接，交接量记为 a_i ： $a_i > 0$ 表示该班组向仓库**存入** a_i 件教具， $a_i < 0$ 表示从仓库**取走** $-a_i$ 件， $a_i = 0$ 表示该班组本次不交接。

仓库有一条硬规矩：任意时刻库存不能为负，取走时若库存不足就无法完成。因此，如果交接是从第 i 个班组才开始的（前面的第 $1 \sim i-1$ 个班组都跳过），老师需要在这场交接开始之前预先备好若干件教具放在仓库里。

老师想做一个盘点：对**每个** $i = 1, 2, \dots, n$ ，分别计算“从第 i 个班组开始交接”所需的最少预存量。预存发生在从第 i 个班组开始的这场交接之前，且第 i 个班组本身的这次交接也要满足库存不为负。

输入格式

从文件 stock.in 中读入数据。

第一行一个正整数 n 。

第二行 n 个整数 $a_1, a_2, \dots, a_n$ ，相邻两个整数之间用一个空格隔开。

输出格式

输出到文件 stock.out 中。

一行 n 个整数，第 i 个整数表示从第 i 个班组开始交接所需的最少预存量，相邻两个整数之间用一个空格隔开。

数据范围

对于所有测试数据，保证： $1 \leq n \leq 10^5$ ， $|a_i| \leq 10^6$ 。

测试点	$n \leq$	特殊性质
1~3	100	无

测试点	$n \leq$	特殊性质
4~6	2500	无
7~10	9×10^4	无
11~16	10^5	无
17~18	10^5	无
19	10^5	A
20	10^5	B

特殊性质 A：每个班组的交接量都为负（即每个班组都在取走教具）。

特殊性质 B：恰好有一个班组的交接量为负，其余班组的交接量都非负。

样例输入 1

```
4
3 -5 2 -4
```

样例输出 1

```
4 7 2 4
```

样例输入 2

```
3
2 3 4
```

样例输出 2

```
0 0 0
```

样例解释

样例 1：从第 1 个班组开始，交接后库存依次为 3, −2, 0, −4，最低跌到 −4，需要预存 4 件。从第 2 个班组开始，库存依次为 −5, −3, −7，最低 −7，需要预存 7 件。从第 3 个班组开始，库存依次为 2, −2，需要预存 2 件。从第 4 个班组开始，库存为 −4，需要预存 4 件。

样例 2：每个班组都在存入，无论从哪个班组开始，库存只增不减，任何一场都不需要预存。

程老师的跳格训练

文件读写

- 输入文件 jump.in
- 输出文件 jump.out

限制

- 1000ms
- 512MB

题目描述

程老师带着学生做跳格训练。训练场上一排格子编号从 0 到 n ，他站在 0 号格，目标是恰好到达 n 号格。每次跳跃，他可以从当前格 pos 跳到 $pos + L$ (L 是跳跃长度, $1 \leq L \leq k$)，不能越过 n 号格。

每个格子 i ($1 \leq i \leq n$) 上写有一个积分 v_i ，落到该格就获得 v_i 分，积分可以为负。此外训练还有一条“节奏奖励”规则：若本次跳跃的长度与上一次跳跃的长度相同，额外获得 b 分；连续多步保持同一长度，则每步都可获得该奖励。第一次跳跃没有上一次，不获得奖励。

程老师想知道：到达 n 号格最多能获得多少总积分？

输入格式

从文件 jump.in 中读入数据。

第一行三个正整数 n, k, b ，分别表示格子数（编号 0 到 n ）、每次跳跃长度上限和节奏奖励分值。

第二行 n 个整数 $v_1, v_2, \dots, v_n$ ，相邻两个整数之间用一个空格隔开。

输出格式

输出到文件 jump.out 中。

一行一个整数，表示能获得的最大总积分。

数据范围

对于所有测试数据，保证： $1 \leq n \leq 10^5$, $1 \leq k \leq 300$, $1 \leq b \leq 10^5$, $|v_i| \leq 10^4$ 。

测试点	$n \leq$	$k \leq$	特殊性质
1~2	8	3	无
3~4	32	5	无
5~8	100	10	无
9~12	10^4	100	无

测试点	$n \leq$	$k \leq$	特殊性质
13~16	10^5	300	无
17~18	10^5	300	无
19	10^5	300	A
20	10^5	300	B

特殊性质 A：所有格子上的积分都不大于 0。

特殊性质 B： $k = n$ （即允许从 0 号格一步跳到 n 号格）。

样例输入 1

```
5 2 3
1 2 -5 4 2
```

样例输出 1

```
16
```

样例输入 2

```
4 2 5
3 -1 -1 3
```

样例输出 2

```
19
```

样例解释

样例 1：最优路线是 $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$ ，每步跳 1 格，落格积分 $1 + 2 - 5 + 4 + 2 = 4$ ；第 2 至第 5 次跳跃的长度都与上一次相同，共得 $4 \times 3 = 12$ 分节奏奖励，总计 16 分。若改走 $0 \rightarrow 2 \rightarrow 4 \rightarrow 5$ ，落格积分 $2 + 4 + 2 = 8$ ，只有 1 次奖励得 3 分，总计 11 分，不如前者。

样例 2：最优路线是 $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4$ ，每步跳 1 格，落格积分 $3 - 1 - 1 + 3 = 4$ ，后三步各得 5 分奖励，共 15 分，总计 19 分。