

# CSP-J 绍兴复赛模拟

## 中小学

### (请选手务必仔细阅读本页内容)

#### 复赛形式及测试

上机编程要求采用 C++ 语言，机房提供的软件：Dev-C++ 5.11，比赛使用操作系统为 Windows 10（以机房实际版本为准）。比赛时是全盘开放，但文件保存位置统一为 E 盘。复赛完毕后，主办方将所有选手源程序收集后统一评测。

#### 复赛注意事项

##### 1. 比赛目录结构示例

选手比赛时，需在本机指定目录下为每题建立对应的题目目录，目录名称与题目名称相同，严格区分大小写。选手根据题目要求，将自己的源程序，放在该题的题目目录下（测试时以源程序为准，不测可执行文件）。

例如：假设题目有 cashier、dune、manhattan 三题。

选手的 ID（编号）是 HL-55，使用的语言是 C++，该选手的选手目录结构及其中的文件如下所示：

```
|---HL-55
|       |---cashier
|       |       |---cashier.cpp
|       |--- dune
|       |       |--- dune.cpp
|       |---manhattan
|       |       |---manhattan.cpp
```

## 2.文件命名

比赛中涉及的所有文件名（包括源程序名、输入文件名、输出文件名），都必须严格按照题目要求命名，严格区分大小写。例如：题目要求输入文件名为 `game.in`、则程序中必须按照该名字打开文件，不能使用 `Game.in`、`GAME.IN` 等名字。

## 3.特别提醒

（1）比赛开始前应先检查本机能否正常使用，如有问题可向监考老师提出。比赛结束后应及时离开机房，但注意不要关机。

（2）选手成绩的评测采用机器评卷的形式，全市集中评测。

```
freopen("grass.in","r",stdin);
```

```
freopen("grass.out","w",stdout);
```

## 一. 题目概况

中文题目名称	收集盒	优秀线	物资分装	作业本
英文题目与子目录名	souji	youxiu	wuzi	zuoye
可执行文件名	souji	youxiu	wuzi	zuoye
输入文件名	souji.in	youxiu.in	wuzi.in	zuoye.in
输出文件名	souji.out	youxiu.out	wuzi.out	zuoye.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

## 二. 提交源程序文件名

对于 C++语言	souji.cpp	youxiu.cpp	wuzi.cpp	zuoye.cpp
----------	-----------	------------	----------	-----------

## 三. 运行内存限制

内存上限	128M	128M	128M	128M
------	------	------	------	------

### 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、特别提醒：评测在 **NOI Linux** 下进行

# 1 收集盒

(souji.cpp)

小明特别喜欢收集卡通卡片，他有一个收集盒专门用来放这些卡片。每张卡片上都印着一个数字，代表这张卡片的“稀有度”（数字越小，稀有度越低）。小明对收集盒有两个固定操作：

当他得到新卡片时，会把卡片放进收集盒（对应操作：输入“ik”，表示放入一张稀有度为 k 的卡片）；

当他想换奖品时，会从收集盒里拿出当前稀有度最低的那张卡片（对应操作：输入“q”，表示取出并查看这张卡片的稀有度）。

请你帮小明记录每次“换奖品”时取出的卡片稀有度是多少。

输入格式

第一行是一个正整数 n，表示小明一共进行了 n 次操作（ $2 \leq n \leq 10000$ ）。

接下来 n 行，每行表示一次操作：

若该行是“ik”（i 和 k 之间用空格隔开），表示放入一张稀有度为 k 的卡片（ $1 \leq k < 2 \times 10^9$ ）；

若该行是单独一个“q”，表示取出当前收集盒中稀有度最低的卡片。

输入数据保证：每次“q”操作时，收集盒里一定有卡片；所有操作中至少有一次“q”操作。

输出格式

每当遇到“q”操作时，输出取出的卡片的稀有度。

样例解释

souji.in	souji.out
5	2
i 5	5
i 2	
q	
i 9	
q	

第 1 次操作 “i 5”：收集盒放入稀有度 5 的卡片，盒中卡片：[5]；

第 2 次操作 “i 2”：收集盒放入稀有度 2 的卡片，盒中卡片：[5,2]；

第 3 次操作 “q”：取出稀有度最低的卡片（2），输出 2，盒中剩余卡片：[5]；

第 4 次操作 “i 9”：收集盒放入稀有度 9 的卡片，盒中卡片：[5,9]；

第 5 次操作 “q”：取出稀有度最低的卡片（5），输出 5，盒中剩余卡片：[9]。

## 2 优秀线

(youxiu.cpp)

绍兴一中一（2）班进行了一次数学小测，全班共有  $n$  名同学。老师批改完试卷后，计划逐一公布每位同学的成绩，并实时计算“优秀线”——即当前已公布成绩中，能达到“优秀”标准的最低成绩。

优秀线的计算规则如下：

班级设定的优秀率为  $w\%$ （即理论上，当前已公布成绩的同学中，前  $w\%$  的同学为优秀）。

若已公布  $p$  个成绩，计划优秀人数为  $\max(1, \lfloor p \times w\% \rfloor)$ （ $\lfloor x \rfloor$  表示对  $x$  向下取整，比如  $\lfloor 3.8 \rfloor = 3$ ； $\max(x, y)$  取  $x$  和  $y$  中较大的数）。

若有同学成绩相同，所有同分的同学都算优秀，因此实际优秀人数可能比计划人数多。此时优秀线是当前所有优秀同学中最低的成绩。

请你帮老师编写一个程序，在每次公布成绩后，即时计算并输出当前的优秀线。

输入

输入文件名为 `youxiu.in`。

第一行有两个整数  $n, w$ 。分别代表班级总人数与优秀率（ $w$  是正整数）。 $n \leq 10^5$

第二行有  $n$  个整数，依次代表逐一公布的同学成绩（成绩均为不超过 600 的非负整数）。 $1 \leq w \leq 99$

输出

输出文件名为 `youxiu.out`。

只有一行，包含  $n$  个非负整数，依次代表每次公布成绩后即时的优秀线。相邻两个整数间用一个空格分隔。

<code>youxiu.in</code>	<code>youxiu.out</code>
10 60	200 300 400 400 400 500 400 400 300
200 300 400 500 600 600 0 300 200 100	300

各测试点的  $n$  如下表：

测试点编号	$n$ =
1~ 3	10
4~ 6	500
7~ 10	2000
11~ 17	$10^4$
18~ 20	105

### 3 物资分装

(wuzi.cpp)

绍兴某中学的班级活动中，需要提前准备一批物资并分装到指定容器中。班级现有  $L$  种物资生成方案，每种方案的特性是：初始有 1 份基础物资，经过 1 小时后，这份物资会按对应方案的规则“增殖”——即第  $i$  种方案 1 小时后将 1 份物资变成  $B_i$  份（ $B_i$  为正整数，增殖规则固定）。

活动要求：选择一种物资生成方案，从 1 份基础物资开始准备，让其按规则增殖。待物资数量达到“刚好能平均分入  $P$  个容器”时，立即停止增殖并开始分装（每个容器内物资份数相同）。

已知容器总数  $P$  的数量较大，但可表示为  $p_1$  的  $p_2$  次方（即  $P=p_1^{p_2}$ ），其中  $p_1$ 、 $p_2$  均为可直接记录的正整数。为了让活动尽早开始，需要找到一种方案，使从开始准备到完成分装的时间（小时数）最短。

输入格式

每组输入数据共三行：

第一行有一个正整数  $L$ ，代表物资生成方案的种类数；

第二行有两个正整数  $p_1$ 、 $p_2$ ，以空格隔开， $p_1^{p_2}$  即表示容器的总数  $P$ ；

第三行有  $L$  个正整数，第  $i$  个数  $B_i$  表示第  $i$  种方案经过 1 小时后的物资数量（即 1 份物资 1 小时后变为  $B_i$  份）。

数据规模

$1 \leq L \leq 10000$

$1 \leq p_1 \leq 30000$ ， $1 \leq p_2 \leq 10000$

$1 \leq B_i \leq 2,000,000,000$

wuzi.in	wuzi.out
1	-1
2 1	
3	
2	2
24 1	
30 12	

样例解释：

样例 1 解释

选择唯一的方案：1 小时后物资数量为 3（无法均分入 2 个容器），2 小时后为 9（仍无法均分）……由于该方案的物资数量始终是 3 的幂次（奇数），而  $P=2$ ，永远无法满足“均分”要求，故输出 -1。

样例 2 解释

第 1 种方案：1 小时后物资 30 份（30 不能被 24 整除），2 小时后 900 份（ $900 \div 24 = 37.5$ ，不整除），3 小时后 27000 份（ $27000 \div 24 = 1125$ ，整除），故需 3 小时；

第 2 种方案：1 小时后物资 12 份（12 不能被 24 整除），2 小时后 144 份（ $144 \div 24 = 6$ ，整除），故需 2 小时。因此最短时间为 2 小时

## 4 作业本

(zuoye.cpp)

上虞中学初一(3)班的王老师要给班上的同学发批改好的作业本。班上共有  $x$  名同学，第  $i$  名同学会在第  $t_i$  分钟走到王老师的讲台前等待领作业本。王老师每次只能同时给一批同学发作业本，发完这批后，需要花  $z$  分钟整理剩下的作业本（整理期间不能发作业本，整理完后可以立刻开始发下一批）。

所有同学都需要领到作业本，王老师可以任意安排每一批发作业本的开始时间。请问：如何安排才能让所有同学的总等待时间最短？（每位同学的等待时间 = 王老师开始发他那批作业本的时刻 - 他走到讲台前的时刻）

输入格式

第一行包含两个正整数  $x, z$ ，以一个空格分开，分别代表班上的同学人数和王老师整理一次作业本需要的时间（分钟）。

第二行包含  $x$  个非负整数，相邻两数之间以一个空格分隔，第  $i$  个整数  $t_i$  代表第  $i$  名同学走到讲台前的时刻。

输出格式

输出一行，一个整数，表示所有同学的总等待时间之和的最小值（单位：分钟）

zuoye.in	zuoye.out
5 1 3 4 4 3 5	0
5 5 11 13 1 5 5	4

样例 1 解释

同学 1 和同学 4 在第 3 分钟走到讲台前，王老师刚好在第 3 分钟开始发第一批，两人等待 0 分钟；发完后花 1 分钟整理，第 4 分钟整理完。

同学 2 和同学 3 在第 4 分钟走到讲台前，王老师立刻开始发第二批，两人等待 0 分钟；整理后第 5 分钟结束。

同学 5 在第 5 分钟走到讲台前，王老师立刻开始发第三批，等待 0 分钟。所有同学都领到作业本，总等待时间为 0。

输入样例

```
5 5
11 13 1 5 5
```

输出样例 2

4

样例 2 解释

王老师这样安排最合理：

同学 3 在第 1 分钟走到讲台前，王老师第 1 分钟开始发第一批，同学 3 等待 0 分钟；发完后整理 5 分钟，第 6 分钟整理完。

同学 4 和同学 5 在第 5 分钟走到讲台前，等王老师整理完，第 6 分钟开始发第二批，两人各等待 1 分钟（ $6-5=1$ ）；整理后第 11 分钟结束。

同学 1 在第 11 分钟走到讲台前，同学 2 在第 13 分钟走到讲台前，王老师第 13 分钟开始发第三批；同学 1 等待 2 分钟（ $13-11=2$ ），同学 2 等待 0 分钟。

总等待时间： $0+1+1+2+0=4$ （这是能达到的最小总等待时间）。

数据规模与约定

对于 100% 的数据， $x \leq 500$ ， $z \leq 100$ ， $0 \leq t_i \leq 4 \times 10^6$ 。