

太戈编程练习题

(请选手务必仔细阅读本页内容)

一、题目概况

中文题目名称	三位分节法	电影节	炼金术	混乱
英文题目与子目录名	digit	movie	alchemy	chaos
可执行文件名	digit	movie	alchemy	chaos
输入文件名	digit.in	movie.in	alchemy.in	chaos.in
输出文件名	digit.out	movie.out	alchemy.out	chaos.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	无	无	无	无
结果比较方式	全文比较, 过滤末行后空行, 不过滤中间行行末空格			
题目类型	传统	传统	传统	传统
运行内存上限	256M	256M	256M	256M

二、提交源程序文件名

对于 C++ 语言	digit.cpp	movie.cpp	alchemy.cpp	chaos.cpp
-----------	-----------	-----------	-------------	-----------

三、编译命令 (不包含任何优化开关)

对于 C++ 语言	<code>g++ -o digit digit -1m</code>	<code>g++ -o movie movie -1m</code>	<code>g++ -o alchemy alchemy.cpp -1m</code>	<code>g++ -o chaos chaos.cpp -1m</code>
-----------	-------------------------------------	-------------------------------------	---	---

1.三位分节法

(digit.cpp/c/pas)

【问题描述】

众所周知，我国读数的习惯是以每四位数为一个数级：如万(数字后面 4 个 0)、亿(数字后面 8 个 0)，这些级分别叫做个级，万级，亿级……而西方的分级则是以三位数为一个数级：如 thousand(千，数字后面 3 个 0)、million(百万，数字后面 6 个 0)、billion(十亿，数字后面 9 个 0)……。现在给你两个整数 a 和 b ，请计算他们的和 $a + b$ ，并把计算的结果以“三位分节法”的形式输出，即从最低位开始每隔三位数加进一个逗号，如果结果少于四位则不需添加。

【输入格式】输入文件 digit.in

输入只有一行包含两个整数，用一个空格隔开，分别表示 a 和 b 。

【输出格式】输出文件 digit.out

输出只有一行， $a + b$ 的结果用三位分节法表达的形式。

【输入输出样例 1】

digit.in	digit.out
-102 -203	-305

【样例说明】结果少于四位，不需要添加逗号。

【输入输出样例 2】

digit.in	digit.out
1000000 -2	999,998

【样例说明】 $1000000 + (-2)$ 的结果为 999998，按题目要求从低位每隔三位数加一个逗号，最高位不足四位则不需要添加逗号，所以该计算结果的三位分节法的形式为 999,998。

【数据规模】

50%数据， $|a|, |b| \leq 500$

100%数据， $|a|, |b| \leq 1000000$

2. 电影节

(movie.cpp)

时空限制：1s/256M，测试数据共 10 组

【问题描述】

电影节上有 n 部电影可以观看，编号从 1 到 n 。第 i 部电影分别从时刻 s_i 开始，时刻 t_i 结束。你正在为自己安排电影观看的日程，所以你要知道 n 个问题的答案：如果看完第 i 部电影，下一部你能最早看的电影是第几部。注意：

1. 要看的两部电影的时间点不可以重叠，哪怕是一个时刻相同都不行。
2. 如果有多部一样早就取编号小的。
3. 如果后面没有电影了就输出 $n+1$ 。

【输入格式】 输入文件 `movie.in`

输入第一行为正整数 n ， n 不超过 200000。第二行为 n 个正整数依次代表每个电影的开始时刻，第三行为 n 个正整数依次代表每个电影的结束时刻，均不超过 10^9 。开始时刻不可能超过结束时刻。

【输出格式】 输出文件 `movie.out`

输出 n 个正整数。

【输入输出样例 1】

movie.in	movie.out
3 2 4 4 3 7 5	2 4 4

【样例说明】 对第 1 部电影(从 2 开始到 3 结束)，看完第 1 部后能看最早的是第 2 部和第 3 部，都从 4 开始，但是第 2 部编号更小，所以输出 2。对第 2 和第 3 部电影，后面没电影，输出 4。

【输入输出样例 2】

movie.in	movie.out
4 1 2 4 5 4 3 7 6	4 3 5 5

【样例说明】 对第 1 部电影(从 1 开始到 4 结束)，看完第 1 部后能看的只有第 4 部。对第 2 部电影，最早能看第 3 部。对于第 3 第 4 部电影，后面没有电影了。

【数据规模】

10%数据， $n \leq 3$

50%数据， $n \leq 10000$

100%数据， $n \leq 200000$

3. 炼金术

(alchemy.cpp/c/pas)

时空限制：1s/256M，测试数据共 15 组

【问题描述】

魔法学院的魔药专业学生这学期的作业是制作魔药，其中一名学生 Kelly 决定用已有的魔药搭配出最多的高级魔药。对于 $1 \leq i \leq N \leq 100$ ，她有 a_i ($0 \leq a_i \leq 10^4$) 单位的魔药 i 。此外，她查阅资料取得了 K ($1 \leq K \leq N$) 个配方，她可以融合若干种魔药各一单位，制造一单位编号大于所有被融合魔药的魔药。另外保证，对于每种魔药，Kelly 最多知道一种制造该魔药的配方。

计算经过一系列转化后，Kelly 可能拥有的魔药 N 的最大单位数。

【输入格式】输入文件 `alchemy.in`

输入的第一行包含 N 。

第二行包含 N 个整数 a_i 。

第三行包含 K 。

以下 K 行每行包含两个整数 L 和 M ($M \geq 1$)，随后是 M 个整数。后 M 个整数表示配方中用于制造一单位魔药 L 所需要被融合的魔药。输入保证 L 大于这 M 个数。

【输出格式】输出文件 `alchemy.out`

输出在应用一系列零次或多次转化后，Kelly 可能拥有的魔药 N 的最大单位数。

【输入输出样例 1】

<code>alchemy.in</code>	<code>alchemy.out</code>
5 2 0 0 1 0 3 5 2 3 4 2 1 1 3 1 2	1

【输入输出样例 1 说明】

在这个例子中，以下是一种最优的转化方式：

1. 将一单位魔药 1 转化为魔药 2。
2. 将一单位魔药 2 转化为魔药 3。
3. 将一单位魔药 3 和魔药 4 转化为魔药 5。

现在 Kelly 还有一单位魔药 1 和一单位魔药 5。她无法再制造更多的魔药 5。

【数据规模】

测试点 2 中，对于 $1 \leq i \leq N$ ，一单位魔药 i 可以被转化为一单位魔药 $i+1$ 。

测试点 3-4 中，每个配方均将一单位的一种魔药转化为另一种魔药。

测试点 5-11 没有额外限制。

4. 混乱

(chaos.cpp)

时空限制：1s/256M，测试数据共 10 组

【问题描述】

有一个字符串长度为 n ，每个字符都是小写字母 a 到 z 中的一个。如果一段连续的子串里不同的字母太多，你会觉得太混乱。你希望找到混乱程度最小的一段连续子串。定义混乱程度为一段连续子串里不同字母的个数除以这段子串的长度。如果有多段子串的混乱程度一样小，选取最长的那段。

【输入格式】输入文件 `chaos.in` 第一行输入正整数 n ，第二行为长度为 n 的字符串。**【输出格式】**输出文件 `chaos.out` 输出两个整数，用斜杠/隔开，第一个整数为子串里不同字母的个数，第二个整数为子串的长度。注意不要约分。**【输入输出样例 1】**

chaos.in	chaos.out
4 abcd	4/4

【样例说明】选取 `abcd`**【输入输出样例 2】**

chaos.in	chaos.out
7 xyhalla	2/4

【样例说明】选取 `alla`**【输入输出样例 3】**

chaos.in	chaos.out
6 bababu	2/5

【样例说明】选取 `babab`**【数据规模】**保证有 20%数据， $n \leq 2000$ 保证有 20%数据，字符串只包含 `a` 和 `b`保证有 100%数据， $n \leq 100000$