

第十届全国青少年信息学奥林匹克联赛复赛试题

(提高组 三小时完成)

津津的储蓄计划

(save.pas/dpr/c/cpp)

【问题描述】

津津的零花钱一直都是自己管理。每个月的月初妈妈给津津 300 元钱，津津会预算这个月的花销，并且总能做到实际花销和预算的相同。

为了让津津学习如何储蓄，妈妈提出，津津可以随时把整百的钱存在她那里，到了年末她会加上 20%还给津津。因此津津制定了一个储蓄计划：每个月的月初，在得到妈妈给的零花钱后，如果她预计到这个月的月末手中还会有多于 100 元或恰好 100 元，她就会把整百的钱存在妈妈那里，剩余的钱留在自己手中。

例如 11 月初津津手中还有 83 元，妈妈给了津津 300 元。津津预计 11 月的花销是 180 元，那么她就会在妈妈那里存 200 元，自己留下 183 元。到了 11 月月末，津津手中会剩下 3 元钱。

津津发现这个储蓄计划的主要风险是，存在妈妈那里的钱在年末之前不能取出。有可能在某个月的月初，津津手中的钱加上这个月妈妈给的钱，不够这个月的原定预算。如果出现这种情况，津津将不得不在这个月省吃俭用，压缩预算。

现在请你根据 2004 年 1 月到 12 月每个月津津的预算，判断会不会出现这种情况。如果不会，计算到 2004 年年末，妈妈将津津平常存的钱加上 20%还给津津之后，津津手中会有多少钱。

【输入文件】

输入文件 save.in 包括 12 行数据，每行包含一个小于 350 的非负整数，分别表示 1 月到 12 月津津的预算。

【输出文件】

输出文件 save.out 包括一行，这一行只包含一个整数，如果储蓄计划实施过程中出现某个月钱不够用的情况，输出 -x，x 表示出现这种情况的第一个月；否则输出到 2004 年年末津津手中会有多少钱。

【样例输入 1】

```
290
230
280
200
300
```

170
340
50
90
80
200
60

【样例输出 1】

-7

【样例输入 2】

290
230
280
200
300
170
330
50
90
80
200
60

【样例输出 2】

1580

合并果子

(fruit.pas/dpr/c/cpp)

【问题描述】

在一个果园里，多多已经将所有的果子打了下来，而且按果子的不同种类分成了不同的堆。多多决定把所有的果子合成一堆。

每一次合并，多多可以把两堆果子合并在一起，消耗的体力等于两堆果子的重量之和。可以看出，所有的果子经过 $n-1$ 次合并之后，就只剩下一堆了。多多在合并果子时总共消耗的体力等于每次合并所耗体力之和。

因为还要花大力气把这些果子搬回家，所以多多在合并果子时要尽可能地节省体力。假定每个果子重量都为 1，并且已知果子的种类数和每种果子的数量，你的任务就是设计出合并的次序方案，使多多耗费的体力最少，并输出这个最小的体力耗费值。

例如有 3 种果子，数目依次为 1，2，9。可以先将 1、2 堆合并，新堆数目为 3，耗费体力为 3。接着，将新堆与原先的第三堆合并，又得到新的堆，数目为 12，耗费体力为

12. 所以多多总共耗费体力=3+12=15。可以证明 15 为最小的体力耗费值。

【输入文件】

输入文件 fruit.in 包括两行，第一行是一个整数 n ($1 \leq n \leq 30000$)，表示果子的种类数。第二行包含 n 个整数，用空格分隔，第 i 个整数 a_i ($1 \leq a_i \leq 20000$) 是第 i 种果子的数目。

【输出文件】

输出文件 fruit.out 包括一行，这一行只包含一个整数，也就是最小的体力耗费值。输入数据保证这个值小于 2^{31} 。

【样例输入】

```
3
1 2 9
```

【样例输出】

```
15
```

【数据规模】

对于 30% 的数据，保证有 $n \leq 100$ ；
对于 50% 的数据，保证有 $n \leq 5000$ ；
对于全部的数据，保证有 $n \leq 30000$ 。

合唱队形

(chorus.pas/dpr/c/cpp)

【问题描述】

N 位同学站成一排，音乐老师要请其中的 $(N-K)$ 位同学出列，使得剩下的 K 位同学排成合唱队形。

合唱队形是指这样的一种队形：设 K 位同学从左到右依次编号为 $1, 2, \dots, K$ ，他们的身高分别为 T_1, T_2, \dots, T_K ，则他们的身高满足 $T_1 < T_2 < \dots < T_i, T_i > T_{i+1} > \dots > T_K$ ($1 \leq i \leq K$)。

你的任务是，已知所有 N 位同学的身高，计算最少需要几位同学出列，可以使得剩下的同学排成合唱队形。

【输入文件】

输入文件 chorus.in 第一行是一个整数 N ($2 \leq N \leq 100$)，表示同学的总数。第二行有 n 个整数，用空格分隔，第 i 个整数 T_i ($130 \leq T_i \leq 230$) 是第 i 位同学的身高（厘米）。

【输出文件】

输出文件 chorus.out 包括一行，这一行只包含一个整数，就是最少需要几位同学出列。

【样例输入】

```
8
186 186 150 200 160 130 197 220
```

【样例输出】

```
4
```

【数据规模】

对于 50% 的数据，保证有 $n \leq 20$ ；

对于全部的数据，保证有 $n \leq 100$ 。

虫食算

(alpha.pas/dpr/c/cpp)

【问题描述】

所谓虫食算，就是原先的算式中有一部分被虫子啃掉了，需要我们根据剩下的数字来判定被啃掉的字母。来看一个简单的例子：

```
43#98650#45
+ 8468#6633
-----
44445506978
```

其中#号代表被虫子啃掉的数字。根据算式，我们很容易判断：第一行的两个数字分别是 5 和 3，第二行的数字是 5。

现在，我们对问题做两个限制：

首先，我们只考虑加法的虫食算。这里的加法是 N 进制加法，算式中三个数都有 N 位，允许有前导的 0。

其次，虫子把所有的数都啃光了，我们只知道哪些数字是相同的。我们将相同的数字用相同的字母表示，不同的数字用不同的字母表示。如果这个算式是 N 进制的，我们就取英文字母表中的前 N 个大写字母来表示这个算式中的 0 到 $N-1$ 这 N 个不同的数字（但是这 N 个字母并不一定顺序地代表 0 到 $N-1$ ）。输入数据保证 N 个字母分别至少出现一次。

```
BADC
+ CBDA
-----
DCCC
```

上面的算式是一个 4 进制的算式。很显然，我们只要让 ABCD 分别代表 0123，便可以让这个式子成立了。你的任务是，对于给定的 N 进制加法算式，求出 N 个不同的字母分别代表的数字，使得该加法算式成立。输入数据保证有且仅有一组解。

【输入文件】

输入文件 alpha.in 包含 4 行。第一行有一个正整数 N ($N \leq 26$)，后面的 3 行每行有一个由大写字母组成的字符串，分别代表两个加数以及和。这 3 个字符串左右两端都没有空格，从高位到低位，并且恰好有 N 位。

【输出文件】

输出文件 alpha.out 包括一行，在这一行中，应当包含唯一的那组解。解是这样表示的：输出 N 个数字，分别表示 A，B，C……所代表的数字，相邻的两个数字用一个空格隔开，不能有多余的空格。

【样例输入】

5

ABCED

BDACE

EBBAA

【样例输出】

1 0 3 4 2

【数据规模】

对于 30%的数据，保证有 $n \leq 10$ ；

对于 50%的数据，保证有 $n \leq 15$ ；

对于全部的数据，保证有 $n \leq 26$ 。