排序

問題敘述

輸入一個正整數 n $(1 \le n \le 100)$，代表有一個陣列 a，包含 n 個整數，且每項絕對值 $\le 2^{30}$，但你並不會知道陣列裡的值是甚麼。

你可以做的操作有加減乘除，所有操作都是建立在採二補數的 32 bit 有號整數上，

另外有 25 個臨時變數能用，分別是 $b \sim z$，初始值都是 0。

請輸出一個整數代表你要做幾次操作，接著有那麼多行字串代表你所做的操作，使得操作後整個陣列 a 會被由小到大排序。

輸出 + c 1 b 代表 c=a[1]+b，第一個字是 $+ - */$ 其中一個，後面的三項可以是變數或陣列的某項，若是變數的某項就輸出一個介於 $1 \sim n$ 的數字，陣列編號從 1 開始，輸出不得包含 $+ - */b \sim z1 \sim n$ 及空白和換行以外的字。

輸入說明

共一個整數 n。

測試資料範圍

- $1 \le n \le 100$。

輸出說明

每筆操作一行，操作第一項是 $+ - */$，第二、三、四項可以是 $b \sim z$ 其中一個字，或是一個 $1 \sim n$ 的數字，四項之間皆以一個空格隔開。

每行皆應該恰好有三個空白，分隔四項，行首行尾皆不得有空白。

最後一個操作後應有換行符號，也就是最後得有一個空行。

總操作次數不得超過 $\lceil 2.376 * 1642.53^{1.64253} \rceil$。(超過代表大於，而非大於等於，$\lceil x \rceil$ 代表第一個大於等於 x 的整數。)

範例測資

範例輸入 1

```
2
```

範例輸出 1

```
4
+ b 1 2
/ c 1 2
* c d c
- 1 b 2
```

範例說明 1

二補數的 32 bit 有號整數，代表的是用 32 個 bit 來表示一個數字，且最高 bit 為 0 代表非負，1 代表是負數，其餘的 31 個 bit 用來儲存數字的大小。若我們規定最低位是第 0 位，最高位是第 31 位，則對於一個非負整數來說，他的值是 $\sum_{i=0}^{30}$ 第 i bit 的值 (只會是 0 或 1) $\times 2^i$，而對於一個負數來說，他的量值相當於全部的 bit 都取反 (0 變 1，1 變 0) 後當作正數看待得到的值再加 1。

或者可以當作 gnu c/c++ 的正常 int，意思相同。

所有溢位皆按照 gnu c/c++，除法向零取整。

範例測資中的操作是

b=a[1]+a[2]

c=0*(a[1]+a[2])

a[1]=b-a[2]

根據我們的感性通靈，我們假裝他是對的。

(注意到當 $a[1] > a[2]$ 時這會獲得 WA，所以這只是一個輸出格式範例而已，輸出這個拿不到分。)