题目名称	结算日	进行一个拆的解	线性方程组
题目类型	传统型	传统型	传统型
目录	debt	split	equation
可执行文件名	debt	split	equation
输入文件名	debt.in	split.in	equation.in
输出文件名	debt.out	split.out	equation.out
每个测试点时限	1.0秒	1.0秒	1.0秒
内存限制	256MiB	256MiB	256MiB
测试点数目	10	10	10
测试点是否等分	是	是	是
提交源代码文件名	debt.cpp	split.cpp	equation.cpp

结算日

题目描述

“不放债不借债”，贝西多么希望自己可以遵循这个忠告。她已经和她的 N 个朋友有了债务关系，或者借债了，或者放债了。她的 N 个朋友依次标号为 $1 \dots N$。

结算日终于来临了。她知道，朋友欠她的钱比她欠朋友的钱多。她的朋友们分布在一条直线上，第 i 头奶牛站的位置距离谷仓 i 米。贝西打算沿着这条直线行走，从欠她钱的奶牛手里收钱回来，并且还钱给她欠钱的奶牛。

- 当她沿直线移动的时候，她可以要求任何欠她钱的奶牛还全部的钱。
- 当她有足够的钱可以还清她的某个债，就可以把钱给对应的奶牛还清她的债。

奶牛 i 欠贝西 D_i 元 $(-1,000 \le D_i \le 1,000)$，负数表示贝西欠奶牛 i 钱。贝西从谷仓出发，位置为 0，初始贝西没有钱。贝西收回她的所有借债，并且还清她的欠债所需行走的最短距离是多少？

注意：她必须在最后一头奶牛所在的位置，完成她的行走。

输入格式

第一行，一个整数 N。

接下来第 $2 \dots N+1$ 行，第 $i+1$ 行包含一个整数 D_i。

输出格式

一个整数，贝西收回借债并且还清欠债，所需要行走的最短距离（单位为米）

样例

输入 #1

```
5
100
-200
250
-200
200
```

输出 #1

```
9
```

提示

样例解释

3 头奶牛欠贝西钱；她欠 2 头奶牛钱。当她完成结算，她将有 150 元。

```
谷仓   100   -200   250  -200   200
 |       |       |      |      |      |
 ***>**+**>*****>**+
                    *             <  贝西有  350元
              -**<***
              *                   <  贝西有  150元
              ***>****>****>**+
                               *  <  贝西有  350
                         -**<***
                         *
                         ***>***  <  贝西结束她的行走，有  150元
```

数据规模

对于 30% 的数据，$1 \le N \le 6$。

对于 60% 的数据，$1 \le N \le 1000$。

对于 100% 的数据，$1 \le N \le 10^5$。

进行一个拆的解

题目背景

三岁的小明非常不喜欢完整的东西，他甚至连序列都要拆掉。

题目描述

给定序列 $a_1 \dots a_n$，小明想要把它拆成两个子段 $[1, l][l+1, n](1 \le l < n)$，即 $a_1 \dots a_l$ 和 $a_{l+1} \dots a_n$。

由于小明强迫症很严重，他不希望对于这两个子段，其中一个是另一个的 **子序列**，换句话说，他不希望其中一个子段可以通过删掉若干（可能为 0）个元素变成另一个。

在父母出门的时候，小明终于找到了把序列拆开的机会！所以，他想知道，是否存在一种拆解的方式满足：任意一个子段都不是另一个子段的子序列。

输入格式

第一行一个整数 T，表示小明总共要拆解 T 个序列。接下来 T 行，每行描述一个序列：

- 每行第一个整数 n，表示序列长度；
- 接下来 n 个整数，依次代表序列中每一个数。

输出格式

输出共 T 行。

对于每轮游戏，若存在满足条件的序列，输出 `YES`，否则输出 `NO`。

样例

输入 #1

```
2
5 1 2 1 2 1
7 1 2 1 1 2 1 0
```

输出 #1

```
NO
YES
```

提示

样例解释

对于第一个序列，所有拆分方式有：

- $\{1\},\{2,1,2,1\}$。
- $\{1,2\},\{1,2,1\}$。
- $\{1,2,1\},\{2,1\}$。
- $\{1,2,1,2\},\{1\}$。

从任何地方拆开都是不合法的——较短的那个序列都是另一个序列的子序列。

对于第二个序列，其中一种合理的拆分方式为 $\{1,2,1,1,2\},\{1,0\}$。

数据规模与约定

本题采用捆绑测试。

- Subtask 0（10 points）：$a_i=0$。
- Subtask 1（20 points）：$n=10$，保证数据随机生成。
- Subtask 2（30 points）：n 为偶数。
- Subtask 3（40 points）：无特殊限制。

对于所有数据，$1\le T\le 10^5$，$2\le n\le 10^5$，$1\le \sum n\le 10^6$，$0\le a_i\le 100$。

线性方程组

题目描述

已知 n 元线性一次方程组。

$$\begin{cases} a_{1,1}x_1 + a_{1,2}x_2 + \cdots + a_{1,n}x_n = b_1 \\ a_{2,1}x_1 + a_{2,2}x_2 + \cdots + a_{2,n}x_n = b_2 \\ \cdots \\ a_{n,1}x_1 + a_{n,2}x_2 + \cdots + a_{n,n}x_n = b_n \end{cases}$$

请根据输入的数据，编程输出方程组的解的情况。

输入格式

第一行输入未知数的个数 n。

接下来 n 行，每行 $n+1$ 个整数，表示每一个方程的系数及方程右边的值。

输出格式

如果有唯一解，则输出解。你的结果被认为正确，当且仅当对于每一个 x_i 而言结果值与标准答案值的绝对误差或者相对误差不超过 0.01。

如果方程组无解输出 -1。

如果有无穷多实数解，输出 0。

样例

输入 #1

```
3
2 -1 1 1
4 1 -1 5
1 1 1 0
```

输出 #1

```
x1=1.00
x2=0.00
x3=-1.00
```

提示

对于 20% 的数据，$1 \le n \le 2$。

对于 100% 的数据，$1 \le n \le 100$。对于 $\forall 1 \le i, j \le n$，有 $|a_{i,j}| \le 100$，$|b_i| \le 300$。

解一个含有 n 个未知数的方程组的通用方法

以 $n=3$ 为例，假设我们有这么一个方程组：

$$\begin{cases}x_1+2x_2+x_3=2 & (1)\\ 2x_1+5x_2+x_3=2 & (2)\\ 3x_1+8x_2+4x_3=8 & (3)\end{cases}$$

用 $(2)-2(1)$，可得：

$$\begin{cases}x_1+2x_2+x_3=2 & (1)\\ 0x_1+x_2-x_3=-2 & (2)\\ 3x_1+8x_2+4x_3=12 & (3)\end{cases}$$

用 $(3)-3(1)$，可得：

$$\begin{cases}x_1+2x_2+x_3=2 & (1)\\ 0x_1+x_2-x_3=-2 & (2)\\ 0x_1+2x_2+x_3=2 & (3)\end{cases}$$

我们取得了一个阶段性胜利：**把第一列的除了 (1) 之外的 x_1 的系数变为了 0**。

用 $(3)-2(2)$，可得：

$$\begin{cases}x_1+2x_2+x_3=2 & (1)\\ 0x_1+x_2-x_3=0 & (2)\\ 0x_1+0x_2+3x_3=6 & (3)\end{cases}$$

我们又取得了一个阶段性胜利：**把第二列的除了 $(1)(2)$ 之外的 x_2 的系数变为了 0**。

容易发现最终**未知数前面的非 0 系数**，形成了一个**倒三角**， x_3 的取值可以通过 (3) 确定；紧接着，x_2 的取值可以通过 (2) 确定；紧接着，x_1 的取值可以通过 (1) 确定。

上述方程组的解即为：

$$\begin{cases}x_1=-4\\ x_2=2\\ x_3=2\end{cases}$$

我们把**某 x_i 前面的系数变成 0 的这个行为**，叫做消元。对 n 元线性方程组，我们可以通过消元使得非 0 系数矩阵形成一个倒三角，再从后往前代出每个未知量的值。

当然，当面临着**有无穷多解**或者**无解**的情况时，会有一些特殊判断，数据保证只有一个测试点是无穷多解，只有一个测试点是无解，以下给出两个例子分别对应无穷多解和无解的情况，请读者自行思考如何判断：

无穷多解

$$\begin{cases}x_1+2x_2-x_3=6\\ 2x_1-x_2+x_3=-1\\ 4x_1+3x_2-x_3=11\end{cases}$$

无解

$$\begin{cases}x_1+2x_2-x_3=6\\ 2x_1-x_2+x_3=-1\\ 4x_1+3x_2-x_3=12\end{cases}$$