

- 注意事项
- plan
 - 题目描述
 - 约束
 - 输入格式
 - 输出格式
 - 样例 #1
 - 样例输入 #1
 - 样例输出 #1
 - 样例 #2
 - 样例输入 #2
 - 样例输出 #2
 - 样例 #3
 - 样例输入 #3
 - 样例输出 #3
- prs
 - 题目描述
 - 约束条件
 - 输入
 - 输出
 - 示例 1
 - 示例 2
- magics
 - 问题陈述
 - 约束条件
 - 输入
 - 输出
 - 示例 1
 - 示例 2
 - 示例 3

注意事项

1、代码必须放在子文件夹内，子文件夹名与题目英文名一致。文件名（包括程序名和输入输出文件名）必须使用英文小写。主文件名使用自己姓名拼音（小写字母）

- 2、C++编译选项：-O2 -std=c++14。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、若无特殊说明，输入文件中同一行内的多个整数、浮点数、字符串等均使用一个空格分隔。若无特殊说明，结果比较方式为忽略行末空格、文末回车后的全文比较

题目名称	plan	prs	magics
题目类型	传统型	传统型	传统型
子文件夹名	plan	prs	magics
提交文件名	plan.cpp	prs.cpp	magics.cpp
输入文件名	plan.in	prs.in	magics.in
输出文件名	plan.out	prs.out	magics.out
时间限制	2s	2s	2s
内存限制	1024MB	1024MB	1024MB
测试点数目	24	14	17
测试点是否等分	是	是	是

编译选项： `--std=c++14 -O2`

\

plan

题目描述

高桥正在种植一棵植物。在发芽时，它的高度为0cm。将发芽的那一天视为第0天，它每天的高度增加,在*i*天的夜晚增加 2^icm ($0\leq i$)。

高桥的身高为*H*cm。

每天早晨，高桥都会测量自己的身高与这棵植物相比。找出第一天，这棵植物在早晨的高度严格高于高桥的身高。

约束

- $1 \leq H \leq 10^9$
- 所有输入值均为整数。

输入格式

H

输出格式

输出一个整数，表示第一天，植物在早晨的高度超过高桥的身高。

样例 #1

样例输入 #1

54

样例输出 #1

6

植物在第1,2,3,4,5,6天的早晨分别为1cm,3cm,7cm,15cm,31cm,63cm。植物在第6天早晨变得比高桥更高，因此输出6。

样例 #2

样例输入 #2

7

样例输出 #2

植物在第3天的早晨高度为7cm，第4天的早晨高度为15cm。植物在第4天的早晨变得比高桥更高，因此输出4。请注意，在第3天的早晨，植物与高桥一样高，但没有更高。

样例 #3

样例输入 #3

```
262144
```

样例输出 #3

```
19
```

\

prs

题目描述

N 名用户们聚集在一起玩 **RPS 2**。第 i 位用户的用户名是 S_i ，他们的评分是 C_i 。

RPS 2 的玩法如下：

- 按照用户名的字典序给用户分配数字 $0, 1, \dots, N-1$ 。
- 设 T 为前 N 位用户的评分之和。编号为 $T \bmod N$ 的用户是赢家。

请输出赢家的用户名。

▼ 什么是字典序？

字典序，简单来说，就是“单词在字典中出现的顺序”。更准确地说，用于确定由小写英文字母组成的两个不同字符串 S 和 T 的顺序的算法如下：

这里，“S 的 i 位”表示为 S_i 。如果 S 在字典序上小于 T ，我们写 $S<T$ ，如果 S 更大，我们写 $S>T$ 。

1. 设 L 为 S 和 T 中较短字符串的长度。检查 S_i 和 T_i 是否匹配。
2. 如果存在 i 使得 $S_i \neq T_i$ 设 j 为最小的这样的 i 。比较 S_j 和 T_j 。如果 S_j 在字母表顺序上小于 T_j ，则 $S<T$ 。否则， $S>T$ 。算法到此结束。
3. 如果不存在 i 使得 $S_i \neq T_i$ ，比较 S 和 T 的长度。如果 S 比 T 短，则 $S<T$ 。如果 S 更长，则 $S>T$ 。算法到此结束。

约束条件

- $1 \leq N \leq 100$
- S_i 是由长度在 3 和 16 之间的小写英文字母组成的字符串。
- S_1, S_2, \dots, S_N 都是不同的。
- $1 \leq C_i \leq 4229$
- C_i 是一个整数。

输入

输入以以下格式从标准输入给出：

$N \ S_1 \ C_1 \ S_2 \ C_2 \ \dots \ S_N \ C_N$

输出

在一行上输出答案。

示例 1

Input#

```
3
takahashi 2
aoki 6
snuke 5
```

Output#

snuke

三位用户的评分总和为 13。按用户名字典序排序后得到 **aoki**、**snuke**、**takahashi**，因此编号为 0 的是 1，**snuke** 是 2，**aoki** 是 $13 \bmod 3 = 1$ 。

由于 $13 \bmod 3 = 1$ ，输出 **snuke**，他被分配到编号 1。

示例 2

Input#

```
3
takahashi 2813
takahashixx 1086
takahashix 4229
```

Output#

takahashix

\

magics

问题陈述

高桥有一些来自卡片游戏"Magics"的卡片。第 i 张卡片将被称为第 i 张卡片。每张卡片都有两个参数：强度和成本。第 i 张卡片的强度为 A_i ，成本为 C_i 。

他不喜欢弱卡片，所以他会将它们丢弃。具体来说，他将重复以下操作，直到无法再执行为止：

- 选择两张卡片 x 和 y ，使得 $A_x > A_y$ 和 $C_x < C_y$ 。丢弃第 y 张卡片。

可以证明，当无法再执行操作时，剩下的卡片集合是唯一确定的。找出这组卡片。

约束条件

- $2 \leq N \leq 2 \times 10^5$
- $1 \leq A_i, C_i \leq 10^9$
- A_1, A_2, \dots, A_N 都是不同的。
- C_1, C_2, \dots, C_N 都是不同的。
- 所有输入值都是整数。

输入

输入以以下格式从标准输入给出： $N \ A_1 \ C_1 \ A_2 \ C_2 \ \dots \ A_N \ C_N$

输出

假设剩下 m 张卡片，卡片编号为 i_1, i_2, \dots, i_m ,按升序打印这些卡片，格式如下：

$m \ i_1, i_2, \dots, i_m$

示例 1

Input#

```
3
2 4
1 1
3 2
```

Output#

```
2
2 3
```

关注卡片1和3，我们有 $A_1 < A_3$ 和 $C_1 > C_3$ ，所以可以丢弃卡片1。

无法再执行其他操作。此时，剩下卡片2和3，所以打印它们。

示例 2

Input#

```
5
1 1
10 2
100 3
1000 4
10000 5
```

Output#

```
5
1 2 3 4 5
```

在这种情况下，没有卡片可以丢弃。

示例 3

Input#

```
6
32 101
65 78
2 29
46 55
103 130
52 40
```

Output#

```
4
2 3 5 6
```