

T1. 水果桶(fruit)

Description

PaperCloud 有一个桶，她会向桶中依次放入 n 个水果，每个水果有属性 a_i, b_i 。

从第二个水果开始，每次放入一个水果后，PaperCloud 都会研究桶中已放入水果的一项指标。具体来说，她要求出从桶中任取两个不同的水果 i, j ，能得到的 $\frac{a_i - a_j}{b_i - b_j} (i \neq j)$ 的最大值。

由于 PaperCloud 有更重要的事情要做，因此她找到了聪明的你帮忙解决这个问题。

Format

Input

第一行一个整数 n ，表示放入水果的个数。

之后 n 行每行两个整数 a_i, b_i ，表示这 n 个水果的两种属性。

Output

共 1 行 $n - 1$ 个实数，其中第 i 个实数表示放入 $i + 1$ 个水果后桶中水果 $\frac{a_i - a_j}{b_i - b_j} (i \neq j)$ 的最大值。

结果保留两位小数即可。

Samples

样例输入 1

```
10
7 7
10 4
7 9
2 2
8 5
2 8
2 6
3 1
3 3
7 10
```

样例输出 1

```
-1.00 0.00 4.00 4.00 5.00 5.00 5.00 7.00 7.00
```

样例输入 2

见下发文件中的 `fruit2.in`，数据规模与性质与前 60% 的测试点相同。

样例输出 2

见下发文件中的 `fruit2.out`，数据规模与性质与前 60% 的测试点相同。

样例输入 3

见下发文件中的 `fruit3.in`，数据规模与性质与另外 20% 的测试点相同。

样例输出 3

见下发文件中的 `fruit3.out`，数据规模与性质与另外 20% 的测试点相同。

Limitation

对于 60% 的数据， $1 \leq n \leq 10^3$

对于另外 20% 的数据， $1 \leq b_i \leq n$ ，即 b_i 为 n 阶全排列。

对于 100% 的数据，保证 b_i 互不相同。

对于 100% 的数据， $1 \leq n \leq 10^5, 1 \leq a_i, b_i \leq 10^8$

Hint

- 题目要求的最大值的几何意义其实是坐标为 (b_i, a_i) 的点两两之间连线斜率的最大值。
- 可以观察到斜率最大值只能出现在横坐标相邻的两个点之间
- 因此可以维护一个 `set<pair<int,int>>`，每新增一个点就查询它相邻的点（因为横坐标互不相同，所以相邻的点左右各只有一个）
- 右侧相邻的点可以用 `it = S.upper_bound(p)` 获取，注意特判 `it == S.end()` 的情形。
- 左侧相邻的点可以用 `it = S.lower_bound(p), --it` 获取，注意特判 `it == S.begin()` 的情形。
- 注意精度问题。

T2. 矩阵乘法(must)

Description

本题输入数据注意使用 `long long` !

EunTilofy 在刚刚结束的课程小测中，因一道矩阵乘法的题计算错误痛失 20 分。于是她决定苦练自己的计算能力，尤其是矩阵乘法。

但因为 EunTilofy 不想花时间自己写两个矩阵，所以她决定用一些特殊的方式生成它们。

她先随手写了一个序列 a ，然后按以下方式生成两个 $n \times n$ 矩阵 A 和 B ：

$A_{ij} = a_i \wedge a_j, B_{ij} = a_i \vee a_j (i, j = 1, \dots, n)$ (这里 \wedge 和 \vee 分别表示按位与和按位或运算)。

然后她对这两个矩阵做矩阵乘法，得到了矩阵 $C = A \times B$ ： $C_{i,j} = \sum_{k=1}^n A_{i,k} B_{k,j} (i, j = 1, \dots, n)$

但她非常怀疑自己的计算能力。她决定让你来帮她验算一下。但由于她不想对比 n^2 个数字，所以她只需要你求出 C 中所有元素的和即可。

由于答案可能非常大，你只需输出答案对 $10^9 + 7$ 取模的值即可。

Format

Input

第一行一个整数 n ，第二行 n 个整数 a_1, \dots, a_n 。

Output

一行一个整数， C 中所有元素的和对 $10^9 + 7$ 取模的值。

Samples

样例输入

```
3
1 2 4
```

样例输出

```
91
```

Limitation

对于20%的数据, $n \leq 300$

对于60%的数据, $n \leq 3000, a_i < 2^{30}$

对于全部数据, $1 \leq n \leq 5 \times 10^5, 0 \leq a_i < 2^{60}$

Hint

- $\sum_{i=1}^n \sum_{j=1}^n C_{i,j} = \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n (a_i \wedge a_k)(a_k \vee a_j) = \sum_{k=1}^n (\sum_{i=1}^n a_i \wedge a_k)(\sum_{j=1}^n a_j \vee a_k)$
- 后两个求和可以拆二进制位计算, 即分别考虑二进制表示下第 $0, 1, \dots, 59$ 位对答案的贡献。

T3. 一〇二四(game)

Description

Lisorun 正在打一款名为《一〇二四》的游戏。

和《2048》不同，在这个游戏中， n 个数字会在一个关卡开始时全部出现，它们从左到右排成一行，第 i 个数字形如 2^{a_i} 。

玩家可以进行的操作是选择**相邻且相同**的两个数字，将它们合并成两数之和。如果玩家将所有数字合并到只剩下一个，那么ta就通过了这个关卡。

Lisorun 向你展示了一局游戏的过程：

开局时出现了 5 个数字：

4 1 1 2 8

她先将第 2, 3 个数字合并，得到：

4 2 2 8

接下来将第 2, 3 个数字合并，得到：

4 4 8

然后将第 1, 2 个数字合并，得到：

8 8

最后将第 1, 2 个数字合并，通过了这个关卡：

16

她玩累了想休息一下，于是她把这个游戏交给你，请你帮她通关剩下的 T 个关卡。

对每个关卡，你需要先判断能否通过；如果能通过，请给出一种通过方法；如果无法通过，请给出最少剩余的数字数。

Format

Input

第一行一个整数 T 表示剩余的关卡数。

对每个关卡，输入 2 行。其中：

- 第一行一个整数 n 表示关卡开始时出现的数字个数；
- 第二行 n 个整数 a_1, a_2, \dots, a_n 表示出现的数字为 $2^{a_1}, 2^{a_2}, \dots, 2^{a_n}$ 。

Output

对于每个关卡，输出 2 行：

- 第一行一个字符串，表示是否能够通过这个关卡。若能够通过，请输出 `jiu shi xun la!`（包含空格和结尾的英文叹号）；若无法通过，请输出 `si le la!`（包含空格和结尾的英文叹号）。
- 若能够通过，请在第二行中用 $n - 1$ 个整数给出一种通关的方法。第 i 个整数 x_i 表示第 i 步操作要合并剩余数字中的第 $x_i, x_i + 1$ 个。
- 若不能通关，请在第二行输出一个整数表示最少剩余的数字数。

Special Judge

本题使用 Special Judge 进行评测，任何合法且正确的合并方案都能够被接受。

如果你无法给出一种合并方案，你也可以在第二行中任意输出不超过 $n - 1$ 个整数，或者什么也不输出。你需要确保每个关卡的输出都是两行，否则将被判定为格式错误，失去这个测试点的全部分数。

如果你正确判断了所有关卡是否能够通关，但给出的方案或最少剩余数字数有误，你也能够获得 70% 的分数。

Samples

样例输入

```
3
5
2 0 0 1 3
6
1 1 4 5 1 4
4
1 1 3 2
```

样例输出

```
jiu shi xun la!
2 2 1 1
si le la!
5
si le la!
3
```

样例 2

见附加文件中 `game2.{in|ans}`，这组数据满足特殊性质 A。

样例 3

见附加文件中 `game3.{in|ans}`，这组数据满足特殊性质 B。

样例 4

见附加文件中 `game4.{in|ans}`。

Limitation

对于全部测试数据： $1 \leq T \leq 5$ ， $1 \leq n \leq 512$ ， $0 \leq a_i \leq 10^3$ 。

测试点编号	$T =$	$n \leq$	特殊性质
1	2	8	
2	2	16	
3	1	128	
4	5	128	A
5	5	128	
6	1	512	
7	5	512	AB
8	5	512	A
9	5	512	B
10	5	512	

特殊性质：

- A：所有关卡均无法通过；
- B：每个关卡中出现的数字只有一种。

校验器

为了方便选手测试，在下发文件中我们下发了 `checker.cpp` 文件，选手可以编译该程序，并使用它校验自己的输出文件。但请注意它与最终评测时所使用的校验器并不完全一致。你也不需要关心其代码的具体内容。

编译命令为：`g++ checker.cpp -o checker -std=c++11`。

`checker` 的使用方式为：`checker <input_file> <output_file> <answer_file>`，参数依次表示输入文件、你的输出文件和答案文件。

你需要保证提供给校验器的输入文件和答案文件（除了关卡有解时的第二行答案可以为任意的内容）是都合法的，否则校验器将无法运行或无法给出正确的评判。

校验器将对每个关卡分别进行评判，若关卡的第二行输出错误，还将给出错误原因；最终将给出你的得分（满分为 100）。

Hint

- 注意到只有相邻的数可以合并，因此合并为一个数之前的状态一定是左边一个区间 $[1, k]$ 和右边一个区间 $[k, n]$ 各合并为一个数且这两个数相等。
- 进而可以设计一个区间dp，用 $f(l, r)$ 和 $g(l, r)$ 分别表示 $[l, r]$ 不能合并为一个数时最少剩余个数和 $[l, r]$ 能合并为一个数时最终这个数的值。

T4. 跑步打卡(run)

Description

答案注意开 `long long` !

Bartheme 正在跑步打卡。她需要经过 n 个点。

Bartheme 有 n 个打卡任务，第 i 个打卡任务需要她从当前的 1 号点出发到达 i 号点。没有公里数的限制。

Bartheme 在选择好路线之后，可以将她将要经过的最长的道路长度折叠成 0（如果有多条也只能折叠一条），但代价是她将要经过的某一条最短道路长度将会加倍。

Bartheme 想知道每个打卡任务的最短长度分别是多少。

Format

Input

第一行两个整数 n, m 。

以下 m 行每行三个整数 x, y, z ，表示从 x 号点到 y 号点有一条长为 z 的双向道路。

Output

一行 n 个整数，表示第 $1 \dots n$ 个打卡任务的最短长度。

Samples

样例 1 输入

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

样例 1 输出

```
0 1 2 2 4
```

样例 2 输入

```
10 15
1 2 115
```

```
2 3 187
1 4 156
2 4 278
3 4 195
3 5 194
4 5 182
3 6 255
2 7 105
3 7 224
5 8 296
4 8 103
5 9 285
5 10 213
7 10 187
```

样例 2 输出

```
0 115 230 156 312 417 210 206 494 325
```

Limitation

对于全部数据， $n \leq 10^5$, $m \leq 2 \times 10^5$, $1 \leq x, y \leq n$, $1 \leq z \leq 10^9$ 。

子任务 1（12 分）： $n \leq 10$, $m \leq 20$;

子任务 2（27 分）： $n \leq 1000$, $m \leq 2000$;

子任务 3（16 分）：所有边权均为 1；

子任务 4（45 分）：无特殊限制。

Hint

- 原题意等价于任选一条边边权变为 0，再任选一条边边权乘 2。
- 这可以使用分层图最短路解决（带有状态标记的最短路）。