

Problem A. SCUPC

Input file: standard input
Output file: standard output
Time limit: 5 seconds
Memory limit: 1024 megabytes

这是全国大学生程序设计竞赛双流区域赛。

小胖有 3 个长度为 n 的 0/1 串，分别为 s_1, s_2, s_3 ，小胖想让这三个字符串中的 1 的尽量重合，也就是想要尽量多的位置 i ，使得 $s_{i,1} + s_{i,2} + s_{i,3} \geq 2$ 。

小胖有超能力，可以在这三个字符串中选择一个，并对其做任意多次循环左移。

小胖想知道， $\sum_{i=0}^{n-1} [s_{i,1} + s_{i,2} + s_{i,3} \geq 2]$ 的最大值。

Input

第一行输入一个整数 t ，表示数据组数。

对于每组数据，第一行先输入一个整数 n ，表示字符串长度。

接下来三行，每一行输入一个 0/1 串，第 i 个 0/1 串代表 s_i 。

$t \leq 10^5, \sum n \leq 10^5$

Output

对于每组数据，输出一个整数，表示 $\sum_{i=0}^{n-1} [s_{i,1} + s_{i,2} + s_{i,3} \geq 2]$ 的最大值。

Example

standard input	standard output
3	1
10	2
0000001000	3
0000000110	
0000000001	
10	
0000001000	
0000010000	
0000001100	
10	
0000000111	
0000000011	
0001000000	

Note

对字符串 s 做一次循环左移: $s_i \rightarrow s_{(i-1+n) \bmod n}$ 。

例如字符串 ‘1234’，一次循环左移后变成 ‘2341’。

Problem B. 切切切

Input file: standard input
 Output file: standard output
 Time limit: 1 second
 Memory limit: 256 megabytes

小胖有一个长度为 $n + 1$ 的数组 f ，数组下标为 $[-\frac{n}{2}, \frac{n}{2}]$ 。

定义一个数组的权值 $val(f) = \sum_{i=0}^{\frac{n}{2}} ||f_i| - |f_{-i}||$ 。

小胖可以将数组 f 切割成任意数量的一堆数组 A ，有以下要求（假设将数组 f 切成了 m 个一维数组 $A_1 \sim A_m$ ）：

1. $\sum_{i=1}^m A_{i,k} = f_k$ 。
2. $\sum_{i=1}^m val(A_i) + (n + 1) \times (m - 1)$ 的值最小。
3. A_i 为整数。

小胖想请你构造出方案。

Input

第一行一个整数 p ，表示 $n + 1$ 的值。

接下来一行，从左到右第 i 个整数，表示 $f_{-\frac{n}{2}+i-1}$ 。

$3 \leq p \leq 2 \times 10^5, 0 \leq |f_i| \leq 10^9$ ，保证 p 为奇数。

Output

第一行两个整数，第一个整数表示 m ，即为所划分的函数个数，第二个整数表示 $\sum_{i=1}^m val(A_i) + (n + 1) \times (m - 1)$ 的最小值。

接下来输出 m 行，每行共 $n + 1$ 个整数，第 i 行的第 k 个数表示 $A_{i,-n/2+k-1}$ 。

Example

standard input	standard output
5	2 7
0 1 3 0 7	3 0 0 0 3
	-3 1 3 0 4

Problem C. 树哈希

Input file: standard input
Output file: standard output
Time limit: 3 seconds
Memory limit: 512 megabytes

时隔两年，小 H 又一次在某场比赛中写了树哈希，他写出了以下的代码：

```
const int N = 1e6+10, B = 233, mod = 998244353;
int hs[N], sz[N];
//hs[X] means the hash value of X; sz[X] means the size of X's subtree
vector<int> tu[N];
//tu[X] is the set of X's son
ll qpow(ll x, int y){
    ll ans = 1;
    while(y){
        if(y&1) ans = ans * x % mod;
        x = x * x % mod; y >>= 1;
    }
    return ans;
}

void dfs(int x){
    hs[x] = sz[x] = 1;
    vector<pair<int, int> > V;
    for(int p: tu[x]){
        dfs(p); V.push_back({sz[p], hs[p]});
        sz[x] += sz[p];
    }
    sort(V.begin(), V.end());
    //sort by size of subtree
    for(auto now: V){
        hs[x] = (hs[x] * qpow(B, now.first) % mod + now.second) % mod;
    }
}
```

事后，小 H 发现他的树哈希非常有意思，于是想要研究它的性质。

如果一棵有根树，每个节点的父亲节点编号都比这个节点大，那么小 H 就会非常喜欢它。

小 H 想知道，对于所有的 n 个节点的 "好树"，他们根节点的哈希值的和是多少。

Input

第一行一个整数 T ，表示数据组数。

之后 T 行每行一个整数，表示树的点数。

$1 \leq n \leq 10^6, 1 \leq T \leq 5 \times 10^5$

Output

输出 T 行，每行一个整数代表这组数据的答案。

输出对 998244353 取模。

Example

standard input	standard output
3	76223160
4	234
2	164636081
5	

Note

A, B 都为 n 个节点的好树, A 与 B 不同当且存在一个节点, 在 A 和 B 两棵树中这个节点的父亲不同。

Problem D. MIS again

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 1024 megabytes

给定一个区间 $[l, r]$ ，从中选择一个最大整数集合 S ， S 中两两数字之和都不是 2 的幂次，令 $f(l, r)$ 表示 S 集合的大小，也就是 $f(l, r) = |S|$ 。

现在小 F 要求出 $[1, n]$ 中所有子区间 f 的和。

形式化的，你需要求 $\sum_{i=1}^n \sum_{j=i}^n f(i, j)$ 。

Input

第一行一个整数 T ，表示数据组数。

之后 T 行，每行一个整数 n 。

$1 \leq T, n \leq 5 \times 10^5$

Output

T 行，每行输出一个整数表示此组数据的答案

Example

standard input	standard output
3	9
3	30
5	170
10	

Note

样例解释：

$$f(1, 1) = |\{1\}| = 1$$

$$f(1, 2) = |\{1, 2\}| = 2$$

$$f(1, 3) = |\{2, 3\}| = 2$$

$$f(2, 2) = |\{2\}| = 1$$

$$f(2, 3) = |\{2, 3\}| = 2$$

$$f(3, 3) = |\{3\}| = 1$$

$$\text{故 } \sum_{i=1}^3 \sum_{j=i}^3 = 1 + 2 + 2 + 1 + 2 + 1 = 9$$

Problem E. query on HEX

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 256 megabytes

小胖有一个长度为 n 的十六进制数字 k ，作为他的手机密码。

小 H 要去破解他的密码。

小 H 每次可以询问小胖 k 在区间 $[l, r]$ ($1 \leq l \leq r \leq n$) 中有多少不同的字符（每个数位看作一个字符），小胖会耐心地做出解答。

但是小胖的忍耐是有限度的，他最多接受 $4n$ 次询问。

小 H 想要通过这 $4n$ 次询问，得到一个可能的密码。

Input

第一行一个整数 T ，表示数据组数。

每组数据第一行会输入一个整数 n ，表示十六进制数字 k 的长度。

$$\sum n \leq 10000$$

Output

(No output)

Interaction Protocol

交互输入样例：

```
1
3
? 1 1
? 1 2
? 1 3
? 2 3
! 122
```

交互输出样例：

```
1
2
2
1
```

Note

每次交互开始时，应该先读入整数 n 。

你需要以 `? l r` 的格式做出一次询问。

之后你需要向标准输入流读入一个整数，表示 k 在 $[l, r]$ 区间内的不同字符个数。

当询问完成后，需要以 $!k$ 的方式输出，其中 k 表示你给出的答案。

注意，合法的密码有可能不是唯一的，但是通过询问，可以获得一个合法密码构成的集合，你需要最小化这个集合的大小，并从此集合随意挑出一个作为你的答案。

16 进制字符位 $[0-9, a-f]$

在每次输入输出后，一定要清空缓冲区，具体的，使用以下方法：

```
fflush(stdout); //if you use printf
```

或者

```
cout.flush(); //if you use cout
```

Problem F. 0problem

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

给定整数 m 和 k ，寻找最小的非负数 n ，使得 $[n+1, 2n]$ 中正好有 m 个整数二进制下 1 的个数为 k 。

Input

第一行一个正整数 T ，表示数据组数。

接下来 T 行，每行两个整数，表示 m 和 k 。

$1 \leq T \leq 10^4, 0 \leq m \leq 10^{18}, 1 \leq k \leq 63$ 。

Output

共 T 行，每行一个整数，表示第 T 组数据的答案，保证答案在 long long 范围内。

若无解输出 -1 。

Example

standard input	standard output
2	6
2 3	8190
12 13	

Note

对于样例 1，满足二进制中恰好有 3 个 1 的数组是 7 和 11，故 $n = 6$ 满足条件。

Problem G. 树的定向

Input file: standard input
Output file: standard output
Time limit: 3 seconds
Memory limit: 256 megabytes

小胖有一棵树。

这棵树上的边都很有个性，每条树边都有一个方向。

这棵树上的点都很有个性，第 i 个点的入度一定是集合 G_i 里面的数字之一。

这棵树上面有一些边，他们的方向还没有被确定，现在小胖需要你帮助他确定这些边的方向，使得这棵树上每个点都满足上述条件。

Input

第一行一个整数 T ，表示 T 组数据。

对于每组数据，第一行一个整数 n ，表示树上点的个数。

接下来 $n - 1$ 行，每行两个正整数 x_i, y_i ，表示树上点 x_i 和点 y_i 有一条边直接相连。保证给出的数据一定是一棵树。

接下来 n 行，每行输入一个 01 串 s_i ，假设与第 i 个点直接相连的有 d_i 条边，那么 s_i 长度为 $d_i + 1$ 。

如果 $s_{i,j} = 1$ ，那么说明 $j \in G_i$ ($0 \leq j \leq d_i$)

接下来一行一个长度为 $n - 1$ 的字符串 t ， $\{0, 1, ?\} \in t_i$ 。其中若 $t_i = 0$ ，那么第 i 条边是 x_i 指向 y_i ；若 $t_i = 1$ ，那么第 i 条边是 y_i 指向 x_i ；若 $t_i = ?$ ，那么第 i 条边的方向未知，需要你去确定。

$1 \leq \sum n \leq 5 \times 10^5$

Output

对于每组询问，输出一行一个字符串 P 。假设 $t_i = ?$ ，若在答案中 x_i 指向 y_i ，那么 $P_i = 0$ ，反之 $P_i = 1$ 。注意会有很多合法的 P ，那么输出字典序最小的一个。若无解输出 -1。

Example

standard input	standard output
2	-1
4	00111
3 1	
1 4	
4 2	
010	
10	
10	
001	
10?	
6	
3 5	
3 4	
6 4	
3 2	
6 1	
10	
11	
0100	
110	
11	
101	
?????	

Note

点的入度：假设边 x_i 指向 y_i ，那么 y_i 的入度就加一。

字典序：假设有两个长度为 n 的字符串 x, y ， x 的字典序小于 y 的字典序当且仅当存在 $i \leq n$ ，有 $\forall j < i, x_j = y_j$ 且 $x_i < y_i$

Problem H. 最大三角形

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

在 x 轴存在两点 $A(x_0, 0)$ 和 $B(x_1, 0)$ ，在 y 轴存在一点 $C(0, y_0)$ ，这三个点组成了一个三角形。

现在需要用一条平行于 x 轴的直线 k 去割这个三角形，假设 k 与线段 AC 交于点 a ，与线段 BC 交于点 b ，假设在线段 AB 上存在一点 c ， a, b, c 三点能够组成一个新的三角形。

直线 k 和点 c 都是未知的，请你求出三角形 abc 面积的最大值。

Input

第一行一个整数 T 表示数据组数。

之后 T 行，每行三个整数，分别代表 x_0, x_1, y_0 。

$T \leq 10^5$

$-10^7 \leq x_0, x_1, y_0 \leq 10^7$ ， x_0, x_1, y_0 都为整数。

Output

对于每组数据，输出一个整数表示所求出的面积，要求与标准答案误差不超过 10^{-6}

Example

standard input	standard output
3	0.6250000000
-3 -2 -5	22.5000000000
-7 11 -10	4.5000000000
-7 2 4	

Problem I. 走向叶子

Input file: standard input
Output file: standard output
Time limit: 3 seconds
Memory limit: 512 megabytes

给出一棵 n 个点的无根树，每个点有点权 A_u 。定义每个点的**度数**为与这个点通过一条边相连的点的个数，定义**叶子**为度数为 1 的点的个数。设给定的树的叶子组成的集合为 S 。

你需要执行 q 次操作，每次操作有两种类型：

- 1 $u\ w$ 依次考虑每个 $v \in S$ ，将 u 到 v 的链上的每个点的点权增加 w ，也就是说 u 的点权会被增加若干次。
- 2 u 查询点 u 的点权。

Input

第一行两个整数 $n, q (2 \leq n, q \leq 5 \times 10^5)$ 。

接下来 $n - 1$ 行，每行两个整数 u, v 表示一条边。

接下来 q 行，每行一个操作，操作的格式如题。保证 $1 \leq u \leq n, 1 \leq x \leq 10^6$ 。

Output

对于每个 2 操作输出一行一个整数表示答案。

Example

standard input	standard output
5 5	14
1 2	38
1 3	
2 4	
2 5	
1 1 4	
1 5 10	
2 4	
1 3 4	
2 5	

Problem J. 虚树

Input file: standard input
Output file: standard output
Time limit: 3 seconds
Memory limit: 256 megabytes

小 H 有有根树一棵树，这棵树有一个特殊性质：每个节点的父亲的编号都比这个节点的编号小。

小 H 不告诉小 G 他的树的具体情况，只告诉小 G 每个节点的度数，小 G 想知道是否真的存在这样一棵树。

Input

第一行输入一个整数 T ，表示数据组数。

之后对于每组数据，第一行输入一个整数 n ，表示树上点的个数。第二行包含 n 个整数，其中第 i 个数表示第 i 个点的度数。

$$T \leq 10^6, 1 \leq \sum n \leq 10^6$$

Output

对于每组数据，输出 "Yes" 表示存在这样的一棵树，"NO" 表示不存在这样的树，每组数据占一行。

Example

standard input	standard output
5	No
10	No
2 1 1 1 1 1 1 1 1 8	No
6	Yes
5 5 4 1 5 3	No
10	
5 1 1 1 1 1 1 1 1 5	
9	
4 2 2 2 1 1 1 2 1	
3	
1 3 3	

Note

树： n 个节点， $n - 1$ 条边的连通图。

节点度数：每个节点所连接的边的条数

Problem K. Valeriepieris 圈

Input file: standard input
Output file: standard output
Time limit: 4 seconds
Memory limit: 256 megabytes

Valeriepieris 圈是一个地理概念，指的是在地球表面绘制的一个最小的圆形区域，该区域内居住着全球超过一半的人口。

小H对这个地理问题很感兴趣他把这个问题简化了一下想请你来解决。在平面上有 N 个点，求能覆盖 N 个点中至少 M 个点($N \geq M$)的最小圆,输出最小的圆的半径。

Input

第一行两个整数 N , M ($0 \leq M \leq N \leq 500$) 。

接下来 N 行每行两个整数 x,y 表示每个点的坐标 ($-10^6 \leq x,y \leq 10^6$)

Output

一行一个数表示答案，你的答案视为正确当且仅当与标准答案相差不超过 10^{-4} 。

Examples

standard input	standard output
5 3 0 0 1 1 2 2 3 3 4 4	1.414213670614117
5 2 -10 -5 -5 2 -8 5 -6 9 -9 -10	2.121320434866902
5 3 -3 9 -1 4 3 8 -2 -3 -8 10	3.308924263478730

Problem L. 极牌确认！

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 512 megabytes

Salt 想要在乌蒙 XD 中获得一块极牌，为此她需要在很多谱面中取得 FC(Full Combo) 的成绩。

在乌蒙 XD 中，我们认为一张谱面是由很多音符组成的，每个音符都需要在一个时刻做固定的操作。有以下几种音符：

- x : **note** 音符，表示在这个时刻必须按到 x 号键，这里 x 一定是 $1 \sim 8$ 的一个整数。
- $x[y]$: **hold** 音符，表示在这个时刻必须按到 x 号键，同时在之后的 y 个时刻内不能松开。假设当前时间是 t ，那么 $[t, t+y-1]$ 时刻都必须按着这个键。
- $x-y[z]$: **slide** 音符，表示在当前时刻必须按到 x 号键，且你需要在 z 个时刻之内到达 y 号键。由于你只需要 Full Combo，假设当前时间是 t ，那么必须在 $[t, t+z-1]$ 之内的任意一个时刻移动到 y 号键。我们保证 $x \neq y$ 。

这些元素之间由 `,` 和 `/` 链接，由 `,` 分隔开的音符表示相邻时刻的音符（一个时刻可能没有音符，例如 `,,`），逗号之间的、由 `/` 分隔开的表示同一个时刻的多个音符（不存在 `/` 左侧或右侧为空），组成了一个（在本题中）合法的谱面。我们认为第一个 `,` 前面的部分是 1 时刻的音符，之后每个 `,` 后的音符的时刻是该 `,` 前的音符的时刻加一。

Salt 发现仅凭自己很难完成某些谱面的 FC，因此她召集了一些旅行伙伴，你可以认为一共有 n 只手可以并行地在屏幕上进行移动。为了保证大家不会在游玩过程中乱套（同时他们也懒得抬手），Salt 做出了下面的约定。

你需要选出尽可能多的音符，把它们分成不超过 n 个子序列，每个子序列表示一只手处理的音符，同时：

- 每只手处理的所有音符的时刻互不相同。
- 每只手处理的音符所组成的序列必须保证下一个处理的音符和上一个音符结束时的位置相邻，且下一个处理的音符的时刻在上一个音符处理的时刻之后。除了 $x, x+1 (1 \leq x \leq 7)$ 相邻外，我们认为 $1, 8$ 也是相邻的，一个键并不和自己相邻。也就是说，这只手在处理完当前音符后且未处理下一个音符前，你可以认为其位置不会改变，而它处理的下一个音符一定和当前的位置相邻。
- 如果一只手处理了 t 时刻的形如 $x[y]$ 的 **hold** 音符，那么下一个音符除了相邻之外，还必须保证其出现的时刻不早于 $t+y$ 。
- 如果一只手处理了 t 时刻的形如 $x-y[z]$ 的 **slide** 音符，那么处理的下一个音符除了和 y 键相邻之外，还必须保证其出现时间在 $[t+1, t+z]$ 之间。

注意，在处理形如 $x-y[z]$ 的 **slide** 音符时你可以认为位置从 x 直接移动到了 y ， x, y 并不需要保证相邻，而下一个处理的音符的开头应当和 y 相邻。

如果可以把所有音符分成不超过 n 个子序列，并且每个子序列都满足这些条件，那么输出一行一个字符串 FC，否则输出 a/b ，其中 a 表示最多多少个音符可以被分成不超过 n 个合法子序列， b 表示总共的音符数量。

Input

第一行一个整数 $n (1 \leq n \leq 8)$ 。

第二行一个字符串，表示一个合法的谱面，不含多余空格。保证同一时刻内的所有音符的位置不同。音符总数不超过 1400，其中出现的数字也不超过 1400，字符串长度不超过 5000。

Output

输出一行，表示答案。

Examples

standard input	standard output
2 1,2,1,2,1,2[2],1,2,3-5[2],6,1	FC
2 ,5,4-5[3],,6[2]/5-2[3]/7/8	4/6
2 5-8[10],4-1[10],,,,,,,4,,1/8,2/7,3/6,4/5,,,1/2	9/13

Problem M. 奇异"博"士

Input file: standard input
Output file: standard output
Time limit: 8 seconds
Memory limit: 256 megabytes

小H的好兄弟小W最近沉迷德州扑克，为了帮他戒除"读博"，小H决定陪他玩上几局，给他上一课。他先定制了一副有标记的特殊扑克和德国进口最新款的隐形眼镜，这样他就能看到的小W的底牌。但是他还需要计算两人在当前局面的胜率来决定接下来是加注，过牌还是弃牌。小H找到了你来帮他实现这个程序。

德州扑克是一种流行的扑克游戏，玩家通过比较手中的牌型来决定胜负。在游戏中，玩家可以使用手中的两张底牌和桌面上的五张公共牌，在七张牌中取出五张牌来形成自己的牌型，取其中牌型最大的5张牌。一开始每位玩家从牌库(一副完整的扑克除去大小王)中抽取两张牌，接下来五张公共牌会随着游戏阶段的进行不断随机从剩余的牌库中抽取并展示给所有玩家，本题要求你实现一个在两人游戏的情况下的胜率计算程序，输入小H和小W的手牌，输入当前阶段已经抽取并展示的公牌，计算两人在当前局面的胜率和平局的概率。

1. 花色的表示

- 黑桃: 用字母"S"表示 (Spade) 。
- 红桃: 用字母"H"表示 (Heart) 。
- 梅花: 用字母"C"表示 (Club) 。
- 方块: 用字母"D"表示 (Diamond) 。

2. 数字和字母的表示

- A (Ace) : 表示1或14。
- 2-9: 直接用数字表示。
- T: 表示10。
- J (Jack) : 表示11。
- Q (Queen) : 表示12。
- K (King) : 表示13。

3. 组合表示

- 9H: 表示红桃9。
- JS: 表示黑桃J。
- QD: 表示方块Q。
- KC: 表示梅花K。
- AS: 表示黑桃A。

牌型定义 (从大到小)

1. 皇家同花顺 (Royal Flush)

- 由同一花色的A、K、Q、J和10这五张牌组成。这是最高级的牌型，无法被其他牌型超越。
- 示例: 黑桃A (AS)、黑桃K (KS)、黑桃Q (QS)、黑桃J (JS)、黑桃10 (TS)

2. 同花顺 (Straight Flush)

- 五张同一花色的顺子牌(参考规则同花和顺子)。
- 示例：方块6 (6D)、方块7 (7D)、方块8 (8D)、方块9 (9D)、方块10 (10D)

3. 四条 (Four of a Kind)

- 四张相同点数的牌加上任意一张其他牌。
- 示例：四张8 (8S、8H、8D、8C) + 一张K (KS)

4. 葫芦 (Full House)

- 三张相同点数的牌加上两张相同点数的牌。
- 示例：三张9 (9S、9H、9D) + 两张5 (5S、5H)

5. 同花 (Flush)

- 五张同一花色但不连续的牌。
- 示例：红桃A (AH)、红桃K (KH)、红桃Q (QH)、红桃J (JH)、红桃9 (9H)

6. 顺子 (Straight)

- 五张连续的牌，但花色可以不同。A既可以作为最高牌 (A - K - Q - J - 10)，也可以作为最低牌 (A - 2 - 3 - 4 - 5)。
- 示例：梅花5 (5C)、方块6 (6D)、红桃7 (7H)、黑桃8 (8S)、红桃9 (9H)

7. 三条 (Three of a Kind)

- 三张相同点数的牌加上两张不同点数的牌。
- 示例：三张8 (8S、8H、8D) + 一张Q (QS) + 一张5 (5C)

8. 两对 (Two Pair)

- 两对相同点数的牌加上一张其他点数的牌。
- 示例：一对K (KS、KH) + 一对8 (8S、8H) + 一张3 (3C)

9. 一对 (One Pair)

- 两张相同点数的牌加上三张不同点数的牌。
- 示例：一对10 (TS、TH) + 一张K (KS) + 一张8 (8C) + 一张3 (3D)

10. 高牌 (High Card)

- 五张牌中没有形成任何上述牌型的组合。

- 示例：一张A（AS）、一张K（KS）、一张Q（QS）、一张J（JS）、一张9（9D）

相同牌型比较规则

1. 同花顺

- 比较最高牌的大小。如果最高牌相同，继续比较第二高的牌，以此类推。

2. 四条

- 先比较四张相同点数的牌的大小。如果四张牌的点数相同，再比较第五张牌（踢脚牌）的大小。

3. 葫芦

- 先比较三张相同点数的牌的大小。如果三张牌的点数相同，再比较两张相同点数的牌的大小。

4. 同花

- 比较五张牌中最高牌的大小。如果最高牌相同，再比较第二高的牌，以此类推。

5. 顺子

- 比较最高牌的大小。如果最高牌相同，继续比较第二高的牌，以此类推。（注意A-2-3-4-5是最小的顺子）

6. 三条

- 先比较三张相同点数的牌的大小。如果三张牌的点数相同，再比较剩下的两张牌中较大的那张，如果还相同，再比较较小的那张。

7. 两对

- 先比较两对中较大的那对的大小。如果较大的那对相同，再比较较小的那对的大小。如果两对都相同，再比较第五张牌的大小。

8. 一对

- 先比较一对的大小。如果一对相同，再比较剩下的三张牌中最大的那张，如果还相同，再比较第二大的那张，以此类推。

9. 高牌

- 比较五张牌中最大的那张。如果最大的牌相同，再比较第二大的牌，以此类推。

如果两个玩家的牌型和所有比较的牌都相同，则平局。

Input

第一行1个字符串 H 表示 小H的手牌

第二行1个字符串 W 表示 小W的手牌

第三行1个字符串 S 表示当前已经抽取的公牌(S 可能为空)

Output

第一行1个浮点数表示小H在后续所有可能情况下的胜率

第二行1个浮点数表示小W在后续所有可能情况下的胜率

第三行1个浮点数表示后续所有可能情况下平局的概率

当你的答案和标准答案误差不超过 $1e-6$ 时认为你的答案正确

Examples

standard input	standard output
6HTS AHAS AD2H3H4H5H	1.0000000000000000 0.0000000000000000 0.0000000000000000
AHAS ACAD	0.021730954316523 0.021730954316523 0.956538091366954
9S9D TSTD TCACAD9C	0.022727272727273 0.977272727272727 0.0000000000000000

Note

样例一：当前牌局已定，只有一种情况，两人都是同花顺但是小H的2-3-4-5-6比小W的A-2-3-4-5大。

样例二：当前牌局在后续所有可能情况下，只有同花能让两人分别获胜其余都是平局。

样例二：当前牌局在后续所有可能情况下，公牌最后一张为9H时小H才能组成四条超越小W，所以胜率为 $1/44$ 。