A: 水题

水题，直接输出即可

B: 动态规划

将买卖零次和一次全部转化为买卖两次，相当于a≤b≤c≤d。题目等价于枚举a≤b≤c≤d，使得(s[b]/s[a])\*(s[d]/s[c])的值最大。

定义dp1[i]表示i及其左边的最小数，dp2[i]表示i及其右边的最大数。那么，s[i]/dp1[i]代表以b=i时，最大的(s[b]/s[a])，dp2[i]/s[i]代表以c=i时，最大的(s[d]/s[c])。在此基础上，定义dp3[i]表示i及其左边的最大(s[b]/s[a])，dp4[i]表示i及其右边的最大(s[d]/s[c])。枚举i，求dp3[i]\*dp[4]的最大值。时间复杂度O(n)。

C: 图论、有向图判环

题目中说明，当且仅当等待出现闭环时会出现死锁，我们可以将每一项进程看作图的一个点，一个等待（T1 W T2）看作一条T1到T2的有向边，则此题就是判断有向图是否有环。

对有向图判环有两种方法，一是拓扑排序，若能通过拓扑排序找到一个序列，则该图无环。还有一种是深搜，当搜索一条路经遇到一个在栈中的节点则有环。

D: 数论、互质数

定义d[i]为b[i]-a[i]+1，如果d[1]~d[n]互质，则一定AC，否则存在无法AC的可能（证明不作详述）。判断互质的方法有两种，一是对任意两个数判断gcd是否为1，时间复杂度O(logn)，二是对每个数进行素因数分解，判断是否出现相同的因数，时间复杂度O(n)，其中d为d[i]的数值范围（约为数量级）。本题还有O(nlogd)的做法，但需要使用Java中的BigInteger类。

定义s=d[1]\*d[2]\*...\*d[n]，则期望为20× = 10s - 10

E: 数据结构 / 思维

连续xor值为1的子区间，即存在奇数个1的子区间。利用线段树维护区间右端点向左延申的包含奇数个1的子区间数目。询问时，利用线段树将区间拆分，考虑每个小区间对答案的贡献，把加起来即可。

本题也有O(n)的做法，统计每个1左边出现的连续的0的数量，记为这个1的权值，对于所有第奇数次出现的1，建立一个权值前缀和数组，对于所有第偶数次出现的1，建立另一个权值前缀和数组。对于每个询问的R，如果该位置是1，判断是第奇数次还是偶数次出现，并在对应的权值前缀和数组中求L到R的区间和。如果该位置是0则找到它前面的1，类比上述操作。该方法需要考虑较多的边界条件。

F: 思维

两种做法，第一，对于每个位置，统计每个字符出现的次数，最后将每个位置出现奇数次的字符拼接起来就可以。第二，将所有字符串进行逐位异或，即可得到答案。

G: 数学、公式推导

H: 数论、莫比乌斯反演

当点不在边缘上时，过该点作三条边的平行线，将三角阵切分为三个平行四边形和三个正三角形。对于n\*m的平行四边形，在顶点能看到的点数等于[1,n]和[1,m]内互质数的对数+2，对于n阶正三角形，可以通过n\*n的平行四边形推导出对应答案，将这6块的答案相加，再减去重复计算的6个点。当点在边或顶点上时，类比上述方法。计算互质数对数可用莫比乌斯反演实现。

I: 字符串匹配、后缀数组

后缀数组，将所有每一行输入的数视为字符串，将所有输入的字符串拼接，并且记录下每一个原字符串的首字符在新字符串中的位置，这样原问题就转化为求两个字符串a,b的最长公共前缀是否为第二个字符串b的长度。

J: 几何

如果该点不在三角形的边上，则该点和三角形任意两个顶点一共可以组成三个三角形，如果这三个三角形的面积和等于原三角形面积，则点在三角形内。