A. Hard Geometry

假设三角形的面积是 S，则外接圆半径 $R=\frac{abc}{4S}$，内切圆半径 $r=\frac{2S}{a+b+c}$。面积有海伦公式 $S=\sqrt{p(p-a)(p-b)(p-c)}$，其中 $p=\frac{1}{2}(a+b+c)$。直角三角形下还有简化 $R=\frac{c}{2}$ 和 $r=\frac{a+b-c}{2}$。坑点有两个，一是输入给的 c 不一定是斜边，需要取最大值；二是输出要注意精度，这个自行百度学习。还有一点就是使用 `std::cout << std::endl` 回车的需要注意，这个操作每次都会清空输出流，常数很大；还有 `std::cin` 在输入量大的时候需要关同步。

B. Tic Tac Toe

本地先预处理出所有棋盘状态对应的最优解，然后下棋的时候直接模拟。要注意交互格式，练习交互题可以在 $\mathrm{Codeforces}$ 的 $\mathrm{promblemset}$ 里找。预处理考虑对期盼状态压缩，每个位置只有三种状态（黑白或空），因此状态数量上限为 3^9。为了方便代码实现，可以分别将双方棋子位置构成的集合用 2^9，这样总量为 4^9，虽然会有很多无效状态，但是这个复杂度是可以接受的。处理出不同状态之间能一步棋走到的关系，然后就是简单的博弈：如果存在一步棋使得我走到的状态是对方的必败态，那么我这个状态就是必胜态；上面不满足的话，如果存在一步棋使得我走到的状态是对方的平局态，那么我这个状态就是平局态；如果我一步棋走到的所有状态都是对方的必胜太，那么当前就是我的必败态。从棋局终止状态慢慢回推即可。

E. Light

连续的极长段落数量等于点亮的数量减去有多少相邻的灯同时被点亮，前者很好算，下面只考虑后者。考虑一个以颜色为节点的图，数列上每一个对相邻的颜色对应着图上的一条边，这个图是可以有重边有自环的。现在问题就变成了要对点黑白染色，并且计算有多少条边的端点都是黑色。首先对于黑点的自环的计数可以单独考虑，这个很简单。对于非自环边，将重边压缩，边权为之前的重数。现在就是要计算端点都是黑色的边的权值之和，图有 $O(m)$ 个点，$O(n)$ 条边。将每个节点根据它的度数与 $\sqrt{m}$ 的大小关系划分为大点和小点。如果当前修改的节点是小点，那么直接暴力枚举它的邻边计算对于答案的修改量；如果是大点，需要在之前每次修改的时候（不论大小点），维护一下每个大点相邻的黑点边的权值之和。最后的复杂度是 $O(m+n\log n+q\sqrt{n})$。

G. Binary Search Tree

动态规划，用 $f(l,r)$ 表示能表示 $\{a_l,\cdots,a_r\}$集合的二叉搜索树的数量，那么就有递推式

$$f(l,r)=\sum_{i=l}^{r}[\,l=i\text{ or }a_{i-1}\neq a_i\,]f(l,i-1)f(i+1,r).$$

初值当 $l\geq r$ 时, $f(l,r)=1$，也就是单个节点的树和空树。复杂度 $O(n^3)$。

H. Number

用 $d(n)$ 表示数字 n 在十进制下的位数，那么就有等式

$$\sum_{i=0}^{9}a_i=\sum_{i=1}^{n}d(i)-d(x).$$

注意到 $1\leq d(x)\leq d(n)$，所以

$$\sum_{i=1}^{n-1}d(i)\leq\sum_{i=0}^{9}a_i<\sum_{i=1}^{n}d(i).$$

这样就可以二分出 n 的值，$\sum_{i=1}^{n}d(i)$ 的计算自行思考。用 $d(n,p)$ 表示数字 n 在十进制下有多少个 p，那么 $\sum_{i=1}^{n}d(i,p)-a_p$ 就表示 x 在十进制下有多少个 p。对于 $p\neq 0$，计算 $\sum_{i=1}^{n}d(i,p)$ 可以考虑有多少个数字个位是 p，有多少个数字十位是 p，最后加起来。$\sum_{i=1}^{n}d(i,0)=\sum_{i=1}^{n}d(i)-\sum_{p=1}^{9}\sum_{i=1}^{n}d(i,p)$。

I. Tetris

模拟爆搜，不用在意复杂度，随便什么姿势都能过，就看你敢不敢写。