int

将每一个浮点数乘上 10^9，然后问题就可以转化成，有多少对数的乘积至少包含 18 个末尾 0。

然后计算每一个数的 2 和 5 的因子数量。

统计一下 2 和 5 因子数量和都不小于 18 的数对数量即可。

math

显然要将所有数从大到小、从高位到低位依次填入，使高位的数尽量大。

可以贪心地选择将当前的数字要填在哪一个中。

假设两个数已填入的前缀分别为 a, b，当前数字为 c。

若将 c 放在 a 后，乘积为 $(10a + c)b$。

若将 c 放在 b 后，乘积为 $(10b + c)a$。

相减得 $c(b - a)$，我们要选择较大的乘积，若 $b > a$ 则将 c 放在 a 后，若 $a > b$ 则将 c 放在 b 后这只是一个感性的分析，但可以启示我们：将数放在当前较小的前缀后

写个暴力拍一拍，发现这样会错，需要加上一个补丁：当前缀相等时，放在可填的剩余位数较少的前缀后这样就可以了，比较大小与相乘的做法参考高精。

game

我们知道两个人选的一定都是一段区间。考虑枚举 Alice 第一个落子的位置 X，那么最终的答案肯定是区间 $[L, R]$，其中 $X \in [L, R]$，$R - L + 1 = \lceil \frac{n}{2} \rceil$。

Bob 会尽量让 Alice 选择最差的区间。考虑最差的区间是否能被 Bob 逼出来。

假设第一次落子在 X，最差答案为 $[L, R]$。假设 Bob 会落子在 Y。我们需要让除了 $[L, R]$ 之外的所有位置到 X 的距离比到 Y 的距离更长，这样才能保证这些位置被 Y 控制。那么 $2L - X$ 就是一个极好的位置，如果 X 往一个方向延伸，我们对应把 Y 往这个方向延伸即可。也就是说，如果第一步下在 X 的位置，之后就会被逼到所有包含了 X 的 $[L, R]$ 中，$A_L + A_{L+1} + \cdots + A_R$ 最小的位置。单调队列维护这个东西即可。

以上描述全部在 $\bmod N$ 意义下进行。

posters

$k = 1$ 时，直接输出唯一一张海报面积即可。

$k = 2$ 时，注意到一定是贴在相对的两个角旁边最优，直接计算面积并。

$k = 3$ 时，我们发现一个性质：最大的海报贴在角落是最优的。那么我们枚举剩下两张海报左上角的坐标，计算面积依然采用容斥原理。通过简单的剪枝或者保证较小的常数均可通过这一档的数据。

$k=4$ 以及 $k=5$ 的随机数据时，最大的海报依然要贴在角落，但是直接枚举每张海报的坐标已经不能忍受了，于是我们考虑如何对他进行优化。不难发现，最优的情况下，每张海报至少有两条边要贴着墙边或者之前某张海报的边缘。证明考虑反证法，假如不是最优，那么朝着某个方向平移到边上必然不会更劣而可能更优，假设不成立。因此每次都要贴着边缘走，然后任意时刻至多 $4k+4$ 条边缘直线，每条直线长度至多 100 还不满，因此如果采用搜索的方式实现就能通过这一档。

$k=5$ 的非随机数据，如果搜索不够优秀，那么你可能会在不优的地方深挖下去导致超时，所以我们需要适当进行更多的剪枝。首先我们需要最优性剪枝，如果当前面积加上后面所有海报面积之和还没有超过当前最优答案，直接剪掉。然后我们用 set 去维护每条边缘，加速查询。枚举和搜索需要注意搜索序，优秀的搜索序可以使最优性剪枝的效果更加显著。当然还有一些技巧，需要根据自己的程序实现进行优化。

综合以上可以获得满分。

ring

考场思路修改打标记，询问直接查询区间内所有点，复杂度 $\mathcal{O}(qn)$。

修改打标记，询问查所有环对这个区间的贡献，需要二分查找和前缀和，复杂度 $\mathcal{O}(mn\log n)$。

两者很不平衡，考虑分别操作，为了方便，我们假设 n,m,q 同数量级。把环大小超过和不超过 B 的环分别称作大环和小环。大环数量不会超过 $\frac{n}{B}$。

对小环的操作：暴力移位，维护区间和。树状数组维护：修改 $\mathcal{O}(B\log n)$，查询 $\mathcal{O}(\log n)$，很不平衡。改为分块：修改 $\mathcal{O}(B)$，查询 $\mathcal{O}(B+\frac{n}{B})$。

对大环的操作：同样维护环上前缀和，修改 $\mathcal{O}(1)$，查询 $\mathcal{O}(\frac{n}{B}\log n)$。$B$ 取 $\mathcal{O}(\sqrt{n\log n})$ 达到最优，复杂度为 $\mathcal{O}(n\sqrt{n\log n})$。

然而复杂度不对。我们发现对大环的操作仍然很不平衡。如果要优化二分查找，需要 $\mathcal{O}(\frac{n^2}{B})$ 的空间。在 32MB 下，这是不现实的。但是我们可以离线！我们对于每个大环，跑所有询问，预处理每个位置的 lower_bound。那么只需要 $\mathcal{O}(n)$ 的空间即可优化这个二分。取时间复杂度降为 $\mathcal{O}(n\sqrt{n})$。（还是可以在线做，对于每个块，维护经历大环的移动后，块内该大环最右边和最左边的点被移动到了什么位置，同样用前缀和即可查询，复杂度 $\mathcal{O}(n\sqrt{n})$）。