A、你好所以，再见

解题思路：观察题干，发现题目主要操作是**区间修改以及区间查询**，所以很自然的就会联想到**线段树**这个数据结构（开`4n`大小）。

细节问题：①、如果直接使用线段树不加以修改也是不能ac的，会超时。所以考虑设置一个**lazy数组模拟懒惰操作**，仅在再次访问到相应节点的时候再把lazy信息传递下去同时更新相应子节点的值。②、此外注意到数据范围$x(|\ x\ |\leq 10^6),(1\leq n,q\leq 10^6)$,所以在多次操作后可能会得到一个很大的值，所以**线段树的数组和记录答案的变量都要采用long long。**

时空复杂度：时间复杂度$O(logn)$。空间复杂度：$O(8n)$

C、Mon panache! I

解题思路：通过阅读题意，发现这是要将一个数组分割为相同长度的多段（最后一段长度不够就到数组末尾即可）。然后各段内部排个序，再将所有段连在一起看是不是一个有序的序列，所以**问题可以简化为，取每一段的最大最小值，**将其与相邻的段的最大或最小值比较即可。**由于没有涉及到元素修改的操作，只涉及到区间查询，所以我们可以选择比线段树查询速度还快的ST表**，在O（1）情况下即可完成查询。

细节问题：lg函数向下取整

```
lg[1]=0;
for (int i=2;i<=n;i++) lg[i]=lg[i>>1]+1;
```

涉及到的大量的计算 2^n 操作都可以用移位来表示，移位要比乘除法更快。

考虑到取值范围，所以初始化f[1000001] [20]={0};g[1000001] [20]={0};来分别记录最大最小值。

复杂度：在O(log n)的预处理时间下，O(1)单次查询

G、逆转摄影

解题思路：考虑到正序模拟的话，还有一个找插入点的操作，所以比较麻烦，所以**这道题可以考虑逆序删除，因为终止状态总是确定的。**我们只需要找到后到达的人的顺序就可以了，同时用链表来维护前驱后继（可以用数组模拟链表的方式或许代码要简洁一点）。

细节问题：①、因为题目的数据范围挺大的，所以**存储答案的变量还是得用long long**才可以。②、因为是用长度为n的数组来记录初始顺序。`idx_1 = (idx_1 - 1 + n) % n;`为了避免`idx_1 - 1`小于0的情况所以每次还要加上n再mod n。

复杂度：使用带有前驱后继的链表，在删除操作复杂度就是O（1）

P、光棱塔

解题思路：当对于一个查询 [l , r]，能够在 O (1) 的时间内把 [l , r]的答案转移到相邻的区间 [l , r − 1] [l , r + 1] ...此时可以考虑使用(普通)莫队算法。而本题就围绕着$f(s_i) = s_i * h(s_i)^2$展开。首先按照普通莫队算法**进行分块，以$\sqrt{n}$为一块。然后排序**，首先对于两个询问区间，若其块号相同（即 l 所在的块相同），那么将其 r 作为关键字从小到大排序；若其块号不同（即 l 所在的块不相同），就将 l 作为关键字从小到大排序。

细节问题：①、为了之后能按照正确顺序输出答案，我们可以采用结构体的方式，在结构体里增加一个int id，最后输出的顺序就按照id顺序排列的。②、由于数据范围较大，还有相乘和平方，所以我们保存答案的变量要用long long③、注意add函数和sub函数的逻辑，莫队其他代码都是模版，此处就要根据具体题意来定义我们的添加，减少函数，并判断好使用add（n--）还是add（--n）

复杂度：普通莫队：$O(m\sqrt{n} + n\sqrt{n})$

R、尽力局

解题思路：三维偏序问题，典型的**CDQ分治问题。**一维逆序对用的是归并排序或树状数组。二维偏序则是先排序一维，然后二维使用类似一维求逆序对的方法。所以三维偏序我们也可以从中获得灵感。**对于X**，首先我们将所有三元组按照x进行排序（x相同就看y.....），确保在一维是有序的；**对于Y，**我们采取分治的思想，将大问题分解为一个个小问题，而对于每一个小问题内部，我们对Y进行排序。相邻两个小区间，左区间的x恒小于等于右区间的x的。我们再在左右区间头部设置一个指针，通过比较二者的y值来判断是否将其插入树状数组（f（x）用于记录1-x有多少个元素）中知道左区间指针到达边界，操作完了再看右区间的元素有多少个符合条件。记得每次操作后都要清理树状数组，以免对下次递归造成影响。

注意问题：相同的三元组，我们应该将其结构体`struct node`
`{`
`    int x,y,z,ans,w;     //w表示相同三元组的个数，A[i].ans：当前点的偏序排名，表示有多少其他点在三维空间中位于当前点的前面。`
`                         //A[i].w：当前点的重复次数，表示有多少个与当前点完全相同的点。`
`};`w进行++操作。

W、孩子与玩具

解题思路：显然是一道并查集的模版题。关于这种可以转化为图是否连通的问题，都可以**考虑使用并查集。**判断

细节问题：在find函数中使用一下路径压缩`return toy[x]=find(toy[x]);`，从而之后查询的复杂度大大减小。

X、红魔族首屈一指の恶魔使

解题思路：括号匹配问题，就是**典型的用栈来解决的问题。**如果读入（，则入栈。读入），则首先判断栈顶元素是不是（，如果是就将栈顶元素出栈，如果不是就将）入栈。

细节问题：①、如果n为0的话，直接输出数字0就可以了。②、栈顶指针为0的时候也可能存着（，所以在输入为）时，还需要确定栈顶指针大小来以此分情况讨论。