A - Coin

签到题。统计1的个数/n即可。注意输出格式为print(“%.2f\n”,ans);

B - Time

简单题。将输入的时间s/86400就是增加的天数。接下来就是求当前日期过若干天后是多少天。注意到最大的输入也不会超过27000天，所以一天一天累加改变日期即可。只需眼判断当月有28、29、31还是31天即可。

C – ID

简单题。将输入的所有数字异或起来（^）就是答案。因为a^a=0 , 0^a=a

也可以用其他方式统计。

D – Game

中等题。实际上难度不大，是一个基础的动态规划问题。f[i]表示到时间i时最大能获得多少分数，f[i]=max(f[j]+w[i][j]) 即枚举最后一次练习开始的时间，取最大值即可。注意处理中间的间断。

E – Mod

中等题。注意到不断取模下去会越来越小，而且一个数模a之后再模b，如果a比b小那么模b就没有意义了。所以对输入的a数组剔除所有没用的数，剩下一个递减的数组。在这个数组上，每次二分找到距离当前x最接近且大于x的数，模上这个数重复这一过程即可。这个过程不会重复很多次。很快这个数就会小于所有的a，这时就获得了答案。

F – Number Game

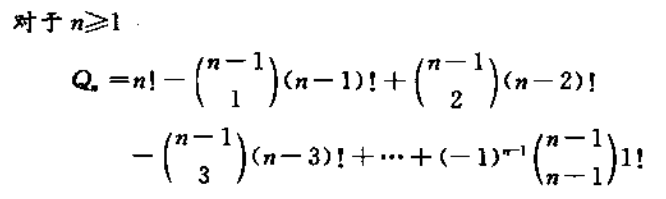
难题。这是一道贪心问题。把数据按ai从大到小排序，ai相同的bi从大到小排序。根据题意a会按照排好的顺序每次拿最大的。那么我们的贪心思路就是每次如果我们可以用当前的bi换下一个会被拿走的bi，从而使得bi的和变大，那我们就交换。这样贪心下去就可以得到答案。具体实现的时候可以用一个堆来动态的获得当前的最大（最小）值。

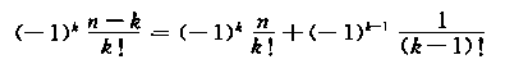
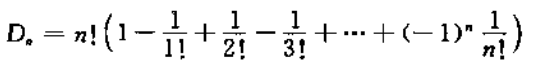
G – OX

中等题。博弈搜索。实际上一共9个位置，每个位置3个状态，一共只有3^9=19683种状态。我们可以暴力枚举出所有的状态，做一个搜索即可。对于当前的一个状态，如果其后继状态里存在先手必败的状态，那么这个状态是先手必胜的；如果其后继状态里没有先手必败的状态但有平局状态，那么这个状态是平局态；如果其后继状态里都是先手必胜态，那么当前状态只能是先手必败了。

H – permutation

中等题。a(n) = n\*a(n-1) + (n-1)\*a(n-2), a(0) = 1, a(1) = 1. a(n) 代表排列 [1,...,n+1] 不含有子串 [k,k+1] 的方案数。所以n 的答案为a(n-1)

实际上，Qn就是我们要求的答案。这个公式比较容易的可以通过容斥原理得到。Qn可以变形为

根据恒等式，我们可以得到，其中Dn是经典的错排数。即。而我们知道Dn是可以递推求出的，，那么将这个递推式带入Qn和Dn的关系就可以得出Qn的递推式。

更加详细的证明可以参考《组合数学》第六章容斥原理及应用6.5节（P113）。

I – Team

中等题。如果直接模拟，主要问题在于撤销操作。撤销操作会撤销其他撤销操作，模拟起来时间复杂度是n^2的。但是如果我们倒着模拟，对于一个操作，如果是一个撤销操作就打个标记，如果一个操作已经被撤销了就什么都不做。这样就不会连锁撤销了。一个人物究竟在不在队伍里取决于最后一次没有被撤销的操作。

J –Emirp

简单题。枚举每个数，并把它翻过来判断是不是素数。最终的答案不会很大，第1000个emirp也就70000左右。

K – Lyne

难题。大模拟。算法上就是一个搜索，考察代码能力。要稍微注意常数。