**1字符串游戏**(strGame.c/cpp/pas)

**前10% 的数据**：直接暴力即可。

**前20% 的数据**：此处即 K很大的情况，思路见正解的该部分处理（只用考虑 K的奇偶）。

**另外 20% 的数据**：此时 K为 1，很显然直接的博弈模型。构建 Trie 树，叶子节点必败。一个非叶节点必败当且仅当其子节点全必胜；一个非叶节点必胜当且仅当其子节点存在一个必败。

**100% 的数据**：

同样构建 Trie 树，由于现在获胜目标变为了最后一轮游戏胜利，所以可以有策略地让某一轮输掉或者胜利。

于是现在一个节点的状态变为了四种，由是否拥有选择胜利本轮的能力以及是否拥有选择输掉本轮的能力区分。

对于叶子节点，只有选择输掉的能力。对于非叶节点，只要有一个子节点不具有选择输掉的能力，该非叶节点便具有选择输掉的能力；只要有一个子节点不具有选择胜利的能力，该非叶节点便具有选择胜利的能力。

也就是说，令 dp[u] 表示节点 u具有的能力，1 表示具有输掉的能力，2 表示具有胜利的能力，3 表示两个能力都有，0 则表示两个能力都没有。那么转移时，dp[u]|= (dp[v]^3)，其中 v为 u的子节点。

不妨令 1 号点为 Trie 树的根节点，那么 Pure 胜利一局游戏的条件便是：根节点处同时具有两种能力（这种情况下，前面n *-* 1 轮一直输，最后一轮胜利即可），或者根节点处仅具有胜利的能力并且 K为奇数（这种情况下，两个人只能轮流获得每一轮的胜利）。

剩余情况显然是 Dirty 获得胜利，例如根节点处仅具有输掉的能力（Pure 只能一直输，显然Dirty 胜利），或者根节点处两者能力都不具有（这种情况下，Dirty 具有控制每一轮游戏胜负的能力，显然 Dirty 胜利）。