为了完成游戏的排名系统，我们需要较为合理的成绩计算方式。

那么考虑会影响分数的数据：

已知的本次拼图游戏的最优解DIFY

玩家解出拼图所用的步数Step

玩家所用的时间Time

由于每一次的游戏都是随机的，那么我们认为最优解DIFY基本可以代表本次游戏的难度，考虑整体趋势，所用Time越大或者所用步数与最优解的差Step-DIFY越大都代表着他本次游戏实际所得分数越低，那么我们反过来设置一个量Bad代表他本次游戏的糟糕度。

根据两个变量Time和Step-DIFY与Bad的关系我们可知

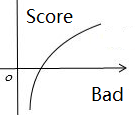
Bad<->Time\*(Step-DIFY)

为了防止出现考虑时间足够长，但所用步数Step与难度DIFY正好相等的情况导致Bad为零，我们令

Bad=Time\*(Step-DIFY)+Time

取糟糕度Bad的对数值得到分数Score，使最终显示的所得分数（实际糟糕程度）在一定程度上减小，减轻程序负担

Score=log(Bad);



Void EndScore(int DIFY,double time){

int Step;

//定义所用步数Step

Double Bad,Score;

//定义游戏糟糕度Bad,最终得分Score

Bad=Time\*(Step-DIFY)+Time;

//根据所推得到糟糕度

Score=log(Bad);

//因为Bad一定大于一，所得对数函数值不会小于1，同时可以缩小所得数据，减轻负担

return Score;

}