**Day1–**枚举、**DFS**

**by zld3794955**

**2018** 年 **2** 月 **5** 日

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 题目名称 | 及格 | 数排列 | 正确答案 | 队列 |
| 目录 | pass | permutation | Accepted | alignment |
| 可执行文件名 | pass | permutation | Accepted | alignment |
| 每个测试点时限 | 1s | 1s | 1s | 1s |
| 内存限制 | 256MB | 256MB | 256MB | 256MB |
| 测试点数量 | 10 | 10 | 10 | 10 |
| 每个测试点分值 | 10 | 10 | 10 | 10 |
| 是否有部分分 | 否 | 否 | 否 | 否 |
| 题目类型 | 传统型 | 传统型 | 传统型 | 传统型 |

注意：预计是**Win7**下**Cena**评测。

评测机性能足够好且评测时开启了**-O2**优化

评测时栈空间设为**16M** ，足以满足正常的函数递归需求。 支持**bits/stdc++.h**

文件输入输出，输入为文件名**.in** ，输出为文件名**.out** 每个人目录下要建子文件夹

及格

【问题描述】

小A这学期上了一门课，这门课分数的评定标准如下：

课程分数=x\*a%+y\*b%+z\*(100-a-b)% ，其中x为平时作业成绩、y为期 中考试成绩、z为期末考试成绩，x,y,z,a,b,100-a-b均是不大于100不小于0的 整数，最终分数如果有小数部分则直接舍去。

现在小A知道了自 己平时作业、 期中考试的成绩以及比例系数a和b， 他想知道为了达到及格（60分）， 自己期末至少要得到多少分，如果小A无 论如何都无法及格，那么就输出”poor little A!”（不含引号）。

【输入格式】

第一行一个正整数T ，表示数据组数。 接下来T行，每行四个非负整数x,y,a,b。

【输出格式】

输出T行，分别表示每组数据中，小A期末要得到多少分，或者”poor little A!”

【样例输入】

3

100 80 20 30

100 100 50 30

100 0 0 50

【样例输出】

32

0

poor little A!

【数据规模】

对于100%的数据：1 ≤ T ≤ 1000, 0 ≤ x,y,a,b ≤ 100, 0 ≤ a+ b ≤ 100。

数排列

【问题描述】

给定一个n个数的序列ai ，其中每项均为0或1。

你的任务是统计符合以下性质的1到n排列（排列中第i项记为pi ）的个 数：

若ai = 0 ，则要求pi 是i的因数，否则要求pi 是i的倍数。

【输入格式】

第一行一个正整数T ，表示数据组数。

接下来T组数据，每组数据第一行一个正整数n表示序列长度，第二 行n个用空格分隔的整数ai。

【输出格式】

对于每组数据输出一行一个正整数表示答案。

【样例输入】

1

3

0 0 0

【样例输出】

1

【数据规模】

对于50%的数据，n ≤ 10。

对于100%的数据，1 ≤ T ≤ 10, 1 ≤ n ≤ 15, 0 ≤ ai ≤ 1。

第 5 页 共[10](#bookmark1)页

正确答案

【问题描述】

n个人参加一场有且仅有m道判断题的考试。

试卷的一份答案是指一个长度为m的字符串，其中每个字符均为Y或N， 且第i个字符表示第i道题的答案。

现在已知每个人的试卷的答案， 以及考试全对的人数k1 及全错的人 数k2 ，试找到该场考试一个可能的正确答案，如果有多种可能，那么就输 出字典序最小的那一个，如果不可能，那么输出-1。

【输入格式】

第一行一个正整数T ，表示数据组数。

接下来T组数据，每组数据第一行有四个非负整数n,m,k1 , k2 ，接下 来n行，每行一个长度为m的由Y或N构成的字符串，表示一个人的答案。

【输出格式】

对每组数据输出一行，表示所求正确答案，或者-1。

【样例输入】

3

2 2 2 0 YY

YY

2 2 0 0 YY

NN

2 2 1 0 YY

YY

【样例输出】

YY NY

-1

【数据规模】

第1,2个测试点满足：1 ≤ n,m ≤ 18。

第3,4个测试点满足：k1 , k2 恰有一个为0。

第5,6个测试点满足：k1 + k2 > 0。 第7,8个测试点满足：k1 = k2 = 0。

对于所有数据均有：1 ≤ T ≤ 100, 1 ≤ n,m ≤ 50, 0 ≤ k1 , k2 , k1 + k2 ≤ n。

队列

【问题描述】

现在有n名玩家围成一圈玩变种版三国杀国战，在这个游戏中一共 有m个势力，编号从1到m ，且每个势力都至少有一名玩家。

第i名玩家有一个自己的初始势力ai ，但为了游戏的公平性，一个势力 的玩家数不能超过X = n/2下取整，当一个势力的玩家数x超过X时，则需 要将这个势力的x − X名玩家变成野心家（野心家与其它野心家和正常势 力都属于不同势力），为了题目方便，不妨假设可以任意选择，在上述过程 后，每名玩家的势力就唯一确定，不会更改了。

当一名玩家死亡之后，其被移出游戏，左右两边的玩家变为相邻。

当一段连续区间内（可以只含有一名玩家） 的玩家都具有相同的势 力A > 0时，我们称该连续段为势力A的一个队列，你的任务就是对每个除 野心家之外的势力A ，计算出在A势力的所有玩家存活的情况下，至少要多 少名不同势力的玩家死亡，才能使A势力的所有玩家形成一个队列，注意 当势力A过大时，除了选择死亡的角色外，你还需要先合理地指定初始势 力A的哪些玩家变成野心家来最小化这个值。

【输入格式】

第一行为一个正整数T ，表示数据组数。

每组数据第一行有两个正整数n,m ，分别表示玩家数和势力数。

接下来一行每行n个不超过m的正整数ai ，按顺时针从某一名玩家开始 给出每名玩家的初始势力。

【输出格式】

对每组数据输出一行m个非负整数，第i个数表示势力i的答案。

【样例输入】

2

8 3

1 2 3 1 2 3 1 2

8 3

1 1 2 1 1 2 1 3

【样例输出】

3 3 2

1 2 0

【样例解释】

第二组数据中，指定第7位的玩家为野心家，可以得到第一种势力的最 优结果1（第3位玩家死亡）。

【数据规模】

对于40%的数据，1 ≤ T ≤ 10,n,m ≤ 300。 对于另外30%的数据，不存在野心家。

对于100%的数据：1 ≤ T ≤ 100, 2 ≤ m ≤ n ≤ 50000 ，数据中n的和不 超过105 ，保证每个势力至少存在一名玩家。

数据规模较大，请使用stdio.h中的scanf和printf函数进行输入输出。