Day 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 题目名称 | 蒙德 | 璃月 | 稻妻 | 须弥 |
| 可执行文件名 | mondstadt | liyue | inazuma | sumeru |
| 输入文件名 | mondstadt.in | liyue.in | inazuma.in | sumeru.in |
| 输出文件名 | mondstadt.out | liyue.out | inazuma.out | sumeru.out |
| 每个测试点时限 | 1秒 | 1秒 | 1秒 | 1秒 |
| 内存限制 | 256MB | 256MB | 256MB | 256MB |
| 测试点数目 | 20 | 20 | 20 | 10 |
| 每个测试点分值 | 5 | 5 | 5 | 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 |
| 是否有样例文件 | 否 | 否 | 否 | 否 |
| 是否有附加文件 | 否 | 否 | 否 | 否 |

提交源程序须加后缀

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 对于C++ 语言 | mondstadt .cpp | liyue.cpp | inazuma.cpp | sumeru.cpp |

编译开关

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 对于C++ 语言 | -O2 -lm | -O2 -lm | -O2 -lm | -O2 -lm |

区间interval国王饮水记drink旷野大计算nodesnodesnodesnodes1.in~nodes10.innodes1.out~nodes10.outnodes1.ans~nodes10.ans

蒙德(Mondstadt)

【问题描述】

在自由的城邦，人们晚上喜欢仰望着星空，尽管那是虚假的。

伟大的占星术士莫娜，绘制了提瓦特天空的无向星图。在星图中，一颗 大的星星指的是星图上一个拥有个点，条边的有向子图（），且所有的边的起点都是一个公共中心。那个公共中心便是星星的中心，而其余节点均为星星的外沿。

莫娜想要计算出无向星图上大小为的星星的数量除以的余数，但是由于输出过于巨大，莫娜只想要知道（）

【输入格式】

输入文件为***mondstadt.in***。

输入文件的第一行包含两个正整数 ，用空格隔开，表示星图节点数和边数。

接下来 行，包含用空格隔开的两个整数 和，为星图中的一条边，保证。

【输出格式与部分分】

输出文件为***mondstadt .out***。

输出仅一行一个整数，表示。

【样例1输入】

3 2

1 2

2 3

【样例1输出】

1

【样例2输入】

4 6

1 2

1 3

1 4

2 3

2 4

3 4

【样例2输出】

8

【样例2说明】

一共有12个大小为2的星和4个大小为3的星，。

【子任务】

所有测试数据的范围和特点如下:

对于的数据有

对于的数据有

对于的数据有 。

对于的数据有 图为 大小的星图

对于的数据有

璃月(Liyue)

【问题描述】

在岩的国度，人们喜欢正方的事物，比如完全平方数。

月海亭的秘书甘雨打算处理今日事务，她将今日的事务从1到n编号，然后打乱，打乱后的第件事编号为。每一次，她会从打乱后的事务中抽出连续的一摞，然后从中选择两个编号之和为完全平方数的事务处理。

甘雨想要知道，假如她抽出的连续的一摞是打乱之后的第到件事务，她有多少种选择两个编号之和为完全平方数的事务的方法。

【输入格式】

输入文件为***liyue.in***。

输入文件的第一行包含一个正整数 。

接下来一行包含个整数，表示打乱后的事务编号。

接下来一行包含一个正整数表示询问次数。

接下来行包含两个正整数，表示甘雨这次抽出的是第到件事务。

【输出格式与部分分】

输出文件为 ***liyue .out***。

输出q行,每行一个整数，该次询问甘雨可选择事务的方案数。

【样例1输入】

8

5 7 4 1 8 6 2 3

10

4 5

2 6

1 8

2 7

4 8

3 8

4 7

1 5

2 5

3 7

【样例1输出】

1

1

5

2

3

3

1

2

1

1

【子任务】

所有测试数据的范围和特点如下:

对于的数据有

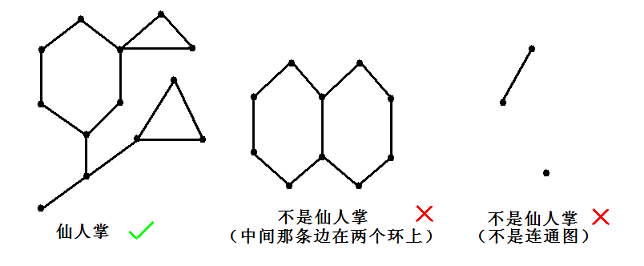
对于的数据有 。

对于额外 的数据有甘雨不会将今日事务打乱

对于的数据有

稻妻(Inazuma)

【问题背景】

在提瓦特，司掌雷电力量的魔神是来自稻妻鸣神岛的*御建鸣神主尊大御所大人*。在雷电的国度，电力供应由一个**仙人掌**形状的输电系统维持。如果一个无向连通图的任意一条边最多属于一个简单环，我们就称之为**仙人掌**。所谓简单环即不经过重复的结点的环。

然而，电线并不是*永恒*的，每根电线都有其寿命。向往永恒的魔神并不能改变万物的寿命，她只能控制每根电线的启用时间。若一根电线的启用时间为，则在时间段内，该电线可以传输电力，其余时间改电线不能传输电力。

整个输电系统只有一个建在号节点的稻妻鸣神岛的发电站，*御建鸣神主尊大御所大人*想知道，这个输电系统最长能令所有节点拥有电力多长时间。

【输入格式】

输入文件为inazuma.in。

输入文件的第一行包含两个正整数 ，用空格隔开，表示节点数和边数。

接下来 行，每行表示一根电线，包含用空格隔开的三个整数 ，和，为电线的两个端点和寿命。

【输出格式】

输出文件为inazuma***.out***。

输出文件只有一行，包含一个正整数，即输电系统正常工作的最长时间。

【样例1输入】

3 3

1 2 1

2 3 1

3 1 2

【样例1输出】

2

【样例1说明】

你可以设置 ，这样所有的节点在时间区间 内都会通过边和边连接，而在时间区间 内则通过边和边连接。

【样例2输入】

5 6

1 2 3

1 2 2

2 3 3

2 4 5

2 5 5

3 4 2

【样例2输出】

5

【子任务】

所有测试数据的范围和特点如下:

对于的数据有

对于的数据有

对于的数据有

对于的数据有

须弥(Sumeru)

【问题背景】

智慧之神纳西妲经常为了封印五百年前的漆黑灾厄，智慧消耗过度（谁是大慈树王？）。现在，她遇到了连智慧之神都无法解决的问题：

对于四个三维向量，是否存在三个非负数，使得。

即,

【输入格式】

输入文件为sumeru.in。

输入文件的第一行包含个正整数 ，表示询问次数。

接下来 行，每行个整数，其中包含，表示。

【输出格式】

输出文件为sumeru***.out***。

输出文件有行，每行输出YES或者NO，表示是否存在满足条件。

【样例1输入】

2

3 4 4 4 3 0 2 3 2 9 10 6

0 3 1 0 1 3 4 0 4 4 1 10

【样例1输出】

YES

NO

【子任务】

所有测试数据的范围和特点如下:

对于的数据有

对于的数据有 线性无关。

对于的数据有