全国青少年信息学奥林匹克联赛

模拟试题

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 中文题目名称 | 数学 | 分数 | 集合 | 城市大脑 |
| 英文题目名称 | math | rating | set | city |
| 输入文件名 | math.in | rating.in | set.in | city.in |
| 输出文件名 | math.out | rating.out | set.out | city.out |
| 测试点时限 | 1秒 | 1秒 | 1秒 | 2秒 |
| 内存限制 | 1024MB | 1024MB | 1024MB | 1024MB |
| 测试点数目 | 20 | 20 | 20 | 20 |
| 测试点分值 | 5 | 5 | 5 | 5 |
| 结果比较方式 | 全文比较（过滤行末空格及文末回车） | | | |
| 题目类型 | 传统 | 传统 | 传统 | 传统 |

提交源程序须加后缀

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 对于C语言 | math.c | rating.c | set.c | city.c |
| 对于C++ 语言 | math.cpp | rating.cpp | set.cpp | city.cpp |

注意：最终测试时，所有编译命令均不打开任何优化开关。

数学

【问题描述】

给定整数，胖教授想将这个数字分成两组，每一组至少有一个数，并且使得两组数字的和的最大公约数最大，请输出最大的最大公约数。

【输入格式】

一行一个整数。

【输出格式】

输出一行一个整数表示答案。

【样例输入1】

6

【样例输出1】

7

【样例输入2】

见下发文件 math/math2.in

【样例输出2】

见下发文件math/math2.ans

【数据规模与约定】

所有测试点的数据规模与约定如下：

|  |  |
| --- | --- |
| 测试点编号 | *n*的规模 |
| 1 | *n* ≤ 100 |
| 2 |
| 3 |
| 4 |
| 5 | *n*≤ 109 |
| 6 |
| 7 |
| 8 |
| 9 |
| 10 |
| 11 |
| 12 |
| 13 |
| 14 |
| 15 |
| 16 |
| 17 |
| 18 |
| 19 |
| 20 |

对于所有数据，。

分数

【问题描述】

在OI大陆，每个OIer都有一个rating。每场比赛以后，参赛者的rating都会发生改变。蜗蜗一共参加了场比赛，所以他的rating一共变动了次。在第场比赛后，他的rating变成了。我们不在意第1场比赛前他的rating是多少。

佳佳是蜗蜗最好的朋友。蜗蜗想让佳佳对他的rating曲线有深刻的了解。每场比赛过后，蜗蜗都可以决定告诉或不告诉佳佳他最新的rating。如果蜗蜗决定告诉佳佳，佳佳会收到蜗蜗最新的rating，否则佳佳会假设蜗蜗的rating仍旧是上次告诉他的时候的样子。蜗蜗想要保证在任何时候，蜗蜗实际的rating和佳佳知道的蜗蜗的rating的差的绝对值不超过。

在第一场比赛过后，蜗蜗必须告诉佳佳他的rating。

请问蜗蜗最少要告诉佳佳几次他的rating？

【输入格式】

第一行两个整数。

第二行个整数。

【输出格式】

输出一行一个整数表示答案。

【样例输入1】

4 0

1 2 2 3

【样例输出1】

3

【样例输入2】

6 5

6 5 0 10 0 10

【样例输出2】

2

【样例输入3】

见下发文件 rating/rating3.in

【样例输出3】

见下发文件rating/rating3.ans

【数据规模与约定】

所有测试点的数据规模与约定如下：

|  |  |
| --- | --- |
| 测试点编号 | *n*的规模 |
| 1 | *n* ≤ 20 |
| 2 |
| 3 |
| 4 |
| 5 | *n* ≤ 5000 |
| 6 |
| 7 |
| 8 |
| 9 |
| 10 |
| 11 |
| 12 |
| 13 |
| 14 |
| 15 |
| 16 |
| 17 |
| 18 |
| 19 |
| 20 |

对于所有数据，。

集合

【问题描述】

对于的子集，它的分数为

1. 初始分数为0；

2. 对于所有的，把加入分数里；

3. 对于所有满足的，如果存在整数满足,从分数里减去；

请求出分数最高的子集的分数是多少。

【输入格式】

输入第一行一个正整数。

接下来一行个整数。

接下来一行个整数。

【输出格式】

输出一行一个数表示答案。

【样例输入1】

4

1 1 1 2

1 1 1 1

【样例输出1】

4

【样例输入2】

4

1 1 1 1

1 1 1 2

【样例输出2】

3

【样例输入3】

见下发文件 set/set3.in

【样例输出3】

见下发文件 set/set3.ans

【数据规模与约定】

所有测试点的数据规模与约定如下：

|  |  |
| --- | --- |
| 测试点编号 | *n*的规模 |
| 1 | *n* ≤ 20 |
| 2 |
| 3 |
| 4 |
| 5 | *n* ≤ 100000 |
| 6 |
| 7 |
| 8 |
| 9 |
| 10 |
| 11 |
| 12 |
| 13 |
| 14 |
| 15 |
| 16 |
| 17 |
| 18 |
| 19 |
| 20 |

对于所有数据，。

城市大脑

【问题描述】

杜老师正在编写杭州城市大脑智能引擎。杭州的道路可以被抽象成为一幅无向图。每条路的初始速度都是 1 m/s。杜老师可以使用 1 块钱让任意一条路的速度提升 1 m/s。如果一条路的速度为m/s，那么我们需要s才能通过这条道路。初始杜老师有块钱，杜老师要把这块钱花在升级这些道路上。 现在有两位选手，一位要从走到，一位要从走到，请问杜老师要怎么花钱，使得两位选手花费的总时间最少？当杜老师花完钱之后，两位选手都会走花费时间最少的路。杜老师花在每条道路上的钱都必须是非负整数。

【输入格式】

第一行三个整数分别表示点数、边数以及杜老师初始有多少钱。

接下来行，每行两个正整数表示边（可以有重边）。

接下来一行四个整数。数据保证、连通。

【输出格式】

一行一个数表示答案。相对误差或绝对误差在内即为正确。

【样例输入1】

6 5 1

1 2

3 2

2 4

4 5

4 6

1 5 3 6

【样例输出1】

5.000000000000

【样例输入2】

1 0 100

1 1 1 1

【样例输出2】

0.000000000000

【样例输入3】

4 2 3

1 2

3 4

1 2 3 4

【样例输出3】

0.833333333333

【样例输入4】

见下发文件 city/city4.in

【样例输出2】

见下发文件 city/city4.ans

【数据规模与约定】

所有测试点的数据规模与约定如下：

|  |  |  |
| --- | --- | --- |
| 测试点编号 | *n,m*的规模 | *k*的规模 |
| 1 | *n,m* ≤ 10 | *k* ≤ 10 |
| 2 |
| 3 |
| 4 |
| 5 | *n,m* ≤ 1,000 | *k* ≤ 109 |
| 6 |
| 7 |
| 8 |
| 9 |
| 10 |
| 11 | *n,m* ≤ 5,000 |
| 12 |
| 13 |
| 14 |
| 15 |
| 16 |
| 17 |
| 18 |
| 19 |
| 20 |

对于所有数据，,,,,,。