全国信息学奥林匹克联赛(模拟赛)

提高组

(请选手务必仔细阅读本页内容)

1. **题目概况**

|  |  |  |  |
| --- | --- | --- | --- |
| 中文题目名称 | 动物园 | 线段计数 | 选择数字 |
| 英文题目名称 | zoo | segment | number |
| 可执行文件名 | zoo | segment | number |
| 输入文件 | zoo.in | segment.in | number.in |
| 输出文件 | zoo.out | segment.out | number.out |
| 每个测试点时限 | 2秒 | 5秒 | 1秒 |
| 测试点数目 | 20 | 10 | 10 |
| 每个测试点分值 | 5 | 10 | 10 |
| 附加样例文件 | 有 | 有 | 有 |
| 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 |
| 运行内存 | 128M | 128M | 64M |

1. **提交源程序文件名**

|  |  |  |  |
| --- | --- | --- | --- |
| 对于C++语言 | zoo.cpp | segment.cpp | number.cpp |
| 对于C语言 | zoo.c | segment.c | number.c |
| 对于pascal语言 | zoo.pas | segment.pas | number.pas |

1. **编译命令（不包含任何优化开关）**

|  |  |  |  |
| --- | --- | --- | --- |
| 对于C++语言 | g++ -o zoo zoo.cpp -lm | g++ -o segment segment.cpp -lm | g++ -o number number.cpp -lm |
| 对于C语言 | gcc -o zoo zoo.c -lm | gcc -o segment segment.c -lm | gcc -o number number.c -lm |
| 对于pascal语言 | fpc zoo.pas | fpc segment.pas | fpc number.pas |

**注意事项：**

1. 文件名（程序名和输入输出文件名）必须使用英文小写。
2. C/C++中函数main()的返回值类型必须是int，程序正常结束返回值必须是0。
3. 在Windows XP的环境下进行最终评测，请C/C++选手使用 %I64d或cin/cout来读入输出64位整数。

**1.动物园**

**(zoo.cpp/c/pas)**

**【问题描述】**

小Z计划带妹子去逛动物园。动物园有个区域，从到编号。第个区域有只动物。动物园里有条道路，道路是无向的，每条道路连接两个区域，且动物园是连通的，每两个区域之间都有路径相连。

小Z为了哄妹子高兴，需要提前规划旅行路线。对于两个区域和间的任意一条简单路径，定义其高兴度为路径上经过的区域中最少的动物数量（包括区域和区域）。小Z自然希望高兴度越高越好，定义、间最大的高兴度为间所有简单路径中高兴度的最大值，记作，现在小Z想知道，对于动物园任意两点的总和是多少。

**【输入格式】**

输入文件名为zoo.in。

输入的第一行为两个整数n和m，第二行包含n个整数，表示每个区域动物的数量。

接下来的m行，每行包含两个数和，表示区域和之间有一条道路。

**【输出格式】**

输出文件名为zoo.out。

输出包含一行，一个整数表示最大高兴度总和，即。

**【输入输出样例】**

|  |  |
| --- | --- |
| zoo.in | zoo.out |
| 4 5  40 20 30 40  1 2  2 3  1 3  2 4  3 4 | 300 |

**【输入输出样例说明】**

总和为

**【数据规模与约定】**

对于10%的测试数据，满足。

对于30%的测试数据，满足。

对于50%的测试数据，满足。

对于100%的测试数据，满足

，保证是连通图且没有重边。

**2．线段计数**

**(segment.cpp/c/pas)**

**【问题描述】**

小Z是一个聪明的孩子，因此经常能收到礼物。

有一天，小Z从小P那里得到了长度为的线段作为礼物，聪明的小Z想将这些线段摆放在数轴上。第次插入操作，将长度为的线段放在数轴上，每一次插入操作后，他都会统计刚刚插入的线段完全覆盖了之前多少条线段。聪明的小Z觉得只插入线段太简单了，于是他又添加了删除的操作，对于删除操作，他都会从数轴上删掉第次插入的线段（线段是相互独立的），现在请你来帮助小Z解决这个问题。

**【输入格式】**

输入文件名为segment.in。

**每个测试点包含多组测试数据。**

输入的第一行为一个正整数，表示测试数据的组数。

每组测试数据包含行。

第一行包含一个正整数，代表操作的次数，接下来行每行包含一个操作，每个操作用两个整数和表示。

如果是0，表示插入操作，小P在位置插入一条线段（对于第次插入的线段，线段将被插入到，长度为）。

如果是1，表示删除操作，小P会删除第次插入操作插入的线段。

**【输出格式】**

输出文件名为segment.out。

对于每组测试数据，首先输出一行‘Case #i:’(不包含引号)，表示第i组测试数据的输出。

接着对于每次的插入操作，按题目要求输出答案。

**【输入输出样例】**

|  |  |
| --- | --- |
| segment.in | segment.out |
| 2  5  0 1  0 3  0 5  1 3  0 2  5  0 1  0 1  1 1  0 2  0 1 | Case #1:  0  0  0  1  Case #2:  0  1  0  2 |

**【输入输出样例说明】**

对于第一组数据：

第一次插入[1,2]线段，没有完全覆盖任何线段，输出0；

第二次插入[3,5]线段，没有完全覆盖任何线段，输出0；

第三次插入[5,8]线段，没有完全覆盖任何线段，输出0；

删除第三次插入的线段；

第四次插入[2,6]线段，完全覆盖第二次插入的线段，输出1。

**【数据规模与约定】**

对于10%的测试数据，满足。

对于30%的测试数据，满足。

对于70%的测试数据，满足。

对于100%的测试数据，满足；

对于插入操作满足；

对于删除操作，保证合法，每个线段最多会被删除1次。

**3.选择数字**

**(number.cpp/c/pas)**

**【问题描述】**

给定N个正整数A1,A2,…,AN，从中选出若干个数，使它们的和为M，求有多少种选择方案。

**【输入格式】**

输入文件名为number.in。

输入第一行包含两个整数和。

第二行包含N个正整数A1,A2,…,AN 。

**【输出格式】**

输出文件名为number.out。

输出只有一行，一个整数表示最后的答案。

**【输入输出样例】**

|  |  |
| --- | --- |
| number.in | number.out |
| 4 4  1 1 2 2 | 3 |

**【输出输入样例说明】**

从1 1 2 2四个数中选取若干个数，使其和为4，共有3种选法分别为(1,1,2)、(1,1,2)、(2,2)。

**【数据规模与约定】**

对于40% 的数据，1 ≤ N ≤ 20, 1 ≤ M ≤ 10

对于100% 的数据，1 ≤ N ≤ 100, 1 ≤ M ≤ 10000，1 ≤ Ai ≤ 1000