测试题

（请选手务必仔细阅读本页内容）

1. 题目概况

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **中文题目名称** | **区间** | **最近公共祖先** | **数列** | **郁闷的出纳员** |
| 英文题目名称 | interval | lca | sequence | unhappy |
| 可执行文件名 | interval | lca | sequence | Unhappy |
| 输入文件名 | interval.in | lca.in | sequence.in | unhappy.in |
| 输出文件名 | interval.out | lca.out | sequence.out | unhappy.out |
| 每个测试点时限 | 1秒 | 1秒 | 1秒 | 1秒 |
| 测试点数目 | 10 | 10 | 10 | 20 |
| 每个测试点分值 | 10 | 10 | 10 | 5 |
| 附加样例文件 | 无 | 无 | 无 | 无 |
| 题目类型 | 传统 | 传统 | 传统 | 传统 |

1. 提交源代码文件名

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 对于pascal语言 | interval.pas | lca.pas | sequence.pas | unhappy.pas |
| 对于C语言 | interval.c | lca.c | sequence.c | unhappy.c |
| 对于C++语言 | interval.cpp | lca.cpp | sequence.cpp | unhappy.cpp |

1. 编译命令（不包含任何优化开关）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 对于pascal语言 | fpc interval.pas | fpc lca.pas | fpc sequence.pas | fpc unhappy.pas |
| 对于C语言 | gcc –o interval  interval.c -lm | gcc –o lca  lca.c -lm | gcc –o sequence  sequence.c -lm | gcc –o unhappy  unhappy.c -lm |
| 对于C++语言 | g++ -o interval  interval.cpp -lm | g++ -o lca  lca.cpp -lm | g++ -o sequence  sequence.cpp -lm | g++ -o unhappy  unhappy.cpp -lm |

1. 允许内存限制

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 内存上限 | 128M | 128M | 128M | 128M |

五、注意事项

1. 每位选手提交一个以自己编号命名的文件夹，其中包含4个源程序和4个子文件夹，名称分别为：interval、lca、sequence、unhappy，每个题目需要上交1个相应的源程序到对应的子文件夹中。多交1个文件夹或文件扣10分，直至考试成绩到0分为止。文件(夹)名称错误不得分。
2. 文件夹名、文件名(程序名和输入输出文件名)必须使用英文小写。
3. C/C++中函数main()的返回值类型必须是int,程序正常结束时的返回值必须是0。
4. 统一评测时采用的机器配置为:windows下lemon评测和全国评测系统下评测。
5. 最终测试时，所有编译命令均不打开任何优化开关。

6、请尽力优化，会收获更多的部分得分。

**第一题： 区间（interval）**

**Description**

**有一些形如[L, R]的区间，你要选出尽可能多的区间，并满足区间两两交集为空（注意[X, X]非空）。**

**Input**

**第一行一个整数 N，**

**接下来 N 行，第 i+1 行有 2 个整数 Li, Ri，描述区间 i。**

**Output**

**第一行一个整数 M，表示最多能选出多少个区间，接下来一行 M 个数，描述你选出的区间。**

**如果有多种选择方案，要求你第二行的输出的字典序最小。**

**Sample Input Sample Output**

**4 2**

**4 9 1 3**

1. **11**
2. **19**
3. **17**

**Hint**

**样例中，你有 3 种选择方案：(1,3)、(2,3)、(1,4)，他们的字典序大小为(1,3) < (1,4) < (2,3). 故你应当选择(1,3)**

**注意输出(1,3)时，输出”3 1”是非法的, 你的输出中编号必须严格递增。**

**Data**

**30%** **N≤2000**

**100% N≤200,000, L,R<109,**

　　**第二题： 最近公共祖先（lca）**

**Description**

**顾名思义.**

**给一棵有根树，以及一些询问，每次询问树上的 2 个节点 A、B，求它们的最近公共祖先。**

**我会用一些方法强制在线。**

**Input**

**第一行一个整数 N，**

**接下来 N 个数，第 i 个数 Fi 表示 i 的父亲是 Fi. 若 Fi = 0,则 i 为树根，**

**接下来一个整数 M，**

**接下来 M 行，每行 2 个整数 A、B，询问节点(A xor LastAns)、(B xor LastAns)的最近公共祖先， 其中 LastAns 为上一个询问的答案，一开始 LastAns = 0。**

**Output**

**对每一个询问输出相应的答案.**

**Sample Input**

**10**

**0 1 2 3 2 4 2 5 4 9**

**10**

**3 9**

**2 7**

**7 8**

**1 1**

**0 6**

**6 11**

**6 3**

**10 7**

**2 15**

**7 7**

**Sample Output**

**3**

**1**

**4**

**5**

**2**

**4**

**2**

**5**

**2**

**5**

**Data**

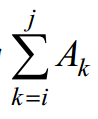
**30%** **n,m≤1000**

**100%** **n,m≤100,000**

**第三题： 数列（sequence）**

**Description**

**有一个长为 n 的数列 A，定义区间[i, j]的权值为：**

****

**现在要你选出k个不相同的区间，使得它们的权值和最大，并满足每个选出的区间长度∈[L, R]。**

**Input**

**第一行四个整数 n， k，L，R 接下来一行n整数，描述 A。**

**Output**

**一行，一个整数描述最大权值和。**

**Sample Input**

**4 3 2 3**

**3**

**2**

-6

**8**

**Sample Output**

**11**

**Hint**

**样例中，选[1, 2]、[2, 4]、[3, 4]是最优的，权值和为 5+4+2=11**

**Data**

|  |  |  |
| --- | --- | --- |
| **数据编号** | **N** | **K** |
| **1** | **10** | **13** |
| **2** | **1000** | **200,000** |
| **3** | **100,000** | **1** |
| **4** | **10000** | **10000** |
| **5** | **500,000** | **10000** |
| **6** | **80000** | **80000** |
| **7** | **100,000** | **100,000** |
| **8** | **100,000** | **500,000** |
| **9** | **500,000** | **500,000** |
| **10** | **500,000** | **500,000** |

**100% -1000≤Ai≤1000.**

**第四题： 郁闷的出纳员（unhappy）**

**Description**

**OIER公司是一家大型专业化软件公司，有着数以万计的员工。作为一名出纳员，我的任务之一便是统计每位员工的工资。这本来是一份不错的工作，但是令人郁闷的是，我们的老板反复无常，经常调整员工的工资。如果他心情好，就可能把每位员工的工资加上一个相同的量。反之，如果心情不好，就可能把他们的工资扣除一个相同的量。我真不知道除了调工资他还做什么其它事情。**

**工资的频繁调整很让员工反感，尤其是集体扣除工资的时候，一旦某位员工发现自己的工资已经低于了合同规定的工资下界，他就会立刻气愤地离开公司，并且再也不会回来了。每位员工的工资下界都是统一规定的。每当一个人离开公司，我就要从电脑中把他的工资档案删去，同样，每当公司招聘了一位新员工，我就得为他新建一个工资档案。**

**老板经常到我这边来询问工资情况，他并不问具体某位员工的工资情况，而是问现在工资第k多的员工拿多少工资。每当这时，我就不得不对数万个员工进行一次漫长的排序，然后告诉他答案。**

**好了，现在你已经对我的工作了解不少了。正如你猜的那样，我想请你编一个工资统计程序。怎么样，不是很困难吧？**

**Input**

**第一行有两个非负整数n和ming, n表示下面有多少条命令, ming表示工资下界。**

**接下来的n行，每行表示一条命令。命令可以是以下四种之一：**

|  |  |  |
| --- | --- | --- |
| **名称** | **格式** | **作用** |
| **I命令** | **I\_k** | **新建一个工资档案，初始工资为k。如果某员工的初始工资低于工资下界，他将立刻离开公司。** |
| **A命令** | **A\_k** | **把每位员工的工资加上k** |
| **S命令** | **S\_k** | **把每位员工的工资扣除k** |
| **F命令** | **F\_k** | **查询第k多的工资** |

**\_（下划线）表示一个空格，I命令、A命令、S命令中的k是一个非负整数，F命令中的k是一个正整数。**

**在初始时，可以认为公司里一个员工也没有。**

**Output**

**输出文件的行数为F命令的条数加一。**

**对于每条F命令，你的程序要输出一行，仅包含一个整数，为当前工资第k多的员工所拿的工资数，如果k大于目前员工的数目，则输出-1。**

**输出文件的最后一行包含一个整数，为离开公司的员工的总数。**

**Sample Input Sample Output**

9 10 10

I 60 20

I 70 -1

S 50 2

F 2

I 30

S 15

A 5

F 1

F 2

**Data**

**I命令的条数不超过100000**

**A命令和S命令的总条数不超过200**

**F命令的条数不超过100000**

**每次工资调整的调整量不超过1000**

**新员工的工资不超过100000**

**30% n<=300**

**100% n<=210000**

**Notice:**

**对于I\_k命令中，若一开始k<ming，则此员工视为未进入公司。**