

乐享未来 NOIP2019 夏令营

模拟测试

(请选手务必仔细阅读本页内容)

一、 题目概况

中文题目名称	不高兴的津津	整理题库	奶牛贝茜的旅行
英文题目与子目录名	unhappy	sortque	cowroute
可执行文件名	unhappy	sortque	cowroute
输入文件名	unhappy	sortque	cowroute
输出文件名	unhappy	sortque	cowroute
每个测试点时限	1 秒		
测试点数目	10	10	10
每个测试点分值	10	10	10
附加样例文件	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		
运行内存上限	128M		

二、 提交源程序文件名

对于 C++语言	unhappy.cpp	sortque.cpp	cowroute.cpp
----------	-------------	-------------	--------------

三、 编译命令

对于 C++语言	<code>g++ -o unhappy unhappy.cpp -lm</code>	<code>g++ -o sortque sortque.cpp -lm</code>	<code>g++ -o cowroute cowroute.cpp -lm</code>
----------	---	---	---

四、 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`, 程序正常结束时的返回值必须是 0。
3. 全国统一评测时采用的机器配置为:
CPU AMD Athlon(tm) II x2 240 processor, 2.8GHz, 内存 4G, 上述时限以此配置为准。
4. 只提供 Linux 格式附加样例文件。
5. 特别提醒: 评测在当前最新公布的 NOI Linux 下进行, 各语言的编译器版本以其为准。

1. 不高兴的津津

(unhappy.cpp)

【题目描述】

津津上初中了。妈妈认为津津应该更加用功学习，所以津津除了上学之外，还要参加妈妈为她报名的各科复习班。另外每周妈妈还会送她去学习朗诵、舞蹈和钢琴。但是津津如果一天上课超过八个小时就会不高兴，而且上得越久就会越不高兴。假设津津不会因为其它事不高兴，并且她的不高兴不会持续到第二天。请你帮忙检查一下津津下周的日程安排，看看下周她会不会不高兴；如果会的话，哪天最不高兴。

【输入输出格式】

输入格式：

输入文件 `unhappy.in` 包括七行数据，分别表示周一到周日的日程安排。每行包括两个小于 10 的非负整数，用空格隔开，分别表示津津在学校上课的时间和妈妈安排她上课的时间。

输出格式：

输出文件 `unhappy.out` 包括一行，这一行只包含一个数字。如果不会不高兴则输出 0，如果会则输出最不高兴的是周几（用 1，2，3，4，5，6，7 分别表示周一，周二，周三，周四，周五，周六，周日）。如果有两天或两天以上不高兴的程度相当，则输出时间最靠前的一天。

【输入输出样例】

输入样例#1：

5 3

6 2

7 2

5 3

5 4

0 4

0 6

输出样例#1：

3

2. 整理题库

(sortque.cpp)

【问题描述】

为了提高大家的程序设计水平，谢老师建立了一个校内题库。题库中共有 n 道题，第 i 道题目的难易程度用 t_i 表示，这 n 道题根据由易到难的顺序已排好。现在老师决定插入 m 道难度为 d_j 的题到题库中，题库中的题仍然按由易到难的顺序排好。

【输入格式】

第一行包含两个用一个空格隔开的 n 和 m 。

第二行包含 n 个用一个空格隔开的正整数 t_j ，表示题库中每道题的难度。

第三行包含 m 个用一个空格隔开的正整数 d_j ，表示待插入的每道题的难度。

$1 \leq t_j \leq 32767, 1 \leq d_j \leq 32767, 1 \leq n \leq 1000, 1 \leq m \leq 1000$ 。

【输出格式】

一行若干个用一个空格隔开的正整数，表示插入后题库的试题难度情况。

【输入样例】

```
5 1
1 1 2 3 7
1
```

【输出样例】

```
1 1 1 2 3 7
```

3. 奶牛贝茜的旅行

(cowroute.cpp)

【题目描述】

奶牛贝茜想到一个更温暖的地方去度过这个寒冷的冬天。不幸的是, 她发现只有一家名叫 AB 的航空公司愿意把票卖给奶牛, 而且这些票的构成很奇怪。AB 有 N 架飞机, 每架都有一个特定飞行路线, 这个飞行路线包含 2 个或更多的城市。例如, 一架飞机的路线可能是从城市 1 开始, 然后飞到城市 6, 再飞到城市 2, 最后飞到城市 8。没有城市会在一条路线上出现多次。如果贝茜决定使用这个路线, 她可以在一条路线的任意一个城市上飞机, 然后在路线上任意一个城市下飞机。她不用一定在第一个城市上飞机, 在最后一个城市下飞机。每条路线会有一个价格, 不管贝茜沿途经过多少城市, 她都要付这么多钱。

贝茜想找到最近的从城市 A 到城市 B 的距离。由于她不想被复杂的行程困惑, 她想只使用一条单独的路线。请帮她决定她最少应该付多少钱。

【输入输出格式】

输入格式:

第 1 行包含 3 个数字 A、B、N。

下面的 2N 行, 描述可用的路线, 每条路线的描述占两行。第一条路线包含路线费用, 自己沿途有多少个城市 (不超过 500 个)。第 2 行包含一个按顺序的城市的列表。

输出格式:

输出贝茜用一条飞行路线从城市 A 飞到城市 B 的最小费用。如果没有这样的路线, 输入 “-1”。

【输入输出样例】

输入样例:

```
1 2 3
3 3
3 2 1
4 4
2 1 4 3
8 5
4 1 7 8 2
```

输出样例:

```
8
```