磁盘文件操作

时间限制： 2.0 秒

空间限制： 512 MiB

相关文件： disk_down.zip (/staticdata/disk_down.zip)

题目背景

小C对计算机运行的原理很感兴趣，经常进行一些研究和实验。

有一天，他在尝试删除一个好几GB大小的文件时，惊奇地发现删除操作几乎在一瞬间就完成了！这让他很是纳闷：如果计算机在每次删除文件时都直接在磁盘上把对应的数据抹掉，不是应该要花挺长时间吗？

于是他找来了小S和小P一起讨论。小S说，或许计算机是一个很“懒”的体系，在删除时不会真的去抹除数据吧？而小P则更见多识广一些，他当即找来了一个号称能“恢复磁盘数据”的软件，当场把小C刚刚删除的文件恢复了！

这让小C有了更强的好奇心，于是他们决定设计一个模型来模拟一个磁盘文件的写入、删除及恢复过程。但是在他们生活的西西艾弗岛上没有合适的条件来运行他们的模型，于是他们联系了带着一台算力超强的电脑来西西艾弗岛旅游的你来帮助他们。

题目描述

在小C、小S和小P设计的模型中，计算机中有 n 段程序（编号为 1-n ），它们共享一块大小为 m 的磁盘空间（编号为 1-m ），磁盘上的每个位置可以写入一个整数。

最初，磁盘上每个位置上的数都是 0 ，并不被任何程序占用。

现在，这 n 段程序同时执行，在某一时刻，某段程序可能对磁盘数据进行读写等操作。

操作共 k 个，按时间先后顺序给出，具体操作如下：

$0\ id\ l\ r\ x$ ：编号为 id 的程序尝试向磁盘空间中 $[l,r]$ 位置上每个位置都写入一个整数 x。

- 操作执行过程中，程序 id 会尝试从最左端 l 开始向右顺次写入数据。
- 对于每个位置，若目前不被任何程序占用，则成功写入整数 x ，并将其视为被程序 id 占用；
- 若该位置目前正被程序 id 自己占用，则这次写入的 x 可以覆盖之前写入的结果，此后该位置仍被程序 id 占用；
- 直到成功向 r 位置写入数据，或遇到第一个正在被其他程序占用的位置为止，此时该操作立刻中断。

$1\ id\ l\ r$ ：程序 id 尝试删除磁盘中 $[l,r]$ 位置上的所有数据。

- 这一操作当且仅当 $[l,r]$ 区间内所有位置都正在被程序 id 占用时才能成功执行。
- 执行效果为将其中所有位置都解除占用，即恢复到可以被任意程序写入的状态。**但为了便于恢复数据，不会立即将全部位置重新覆盖成 0 **。
- 否则，认为此操作执行失败，不进行任何修改。

$2\ id\ l\ r$ ：程序 id 尝试恢复磁盘中 $[l,r]$ 位置上的所有数据。

- 这一操作当且仅当 $[l,r]$ 区间内所有位置都未被占用，且**上一次被占用是被程序 id 占用**时才能成功执行
- 执行效果为将其中所有位置恢复为**被程序 id 占用**的状态，同时由于之前删除操作并未改变其存储的值，因此本次操作也不需要改变每个位置上的值。
- 否则，认为此操作执行失败，不进行任何修改。

$3\ p$：尝试读取磁盘中 p 位置的数据，返回结果为两个整数。

- 如果该位置当前正被程序 id 占用且存储的值为 p，返回结果为 $id\ p$。
- 如果该位置当前没有被任何程序占用，返回 $0\ 0$。

你需要实现一个程序，帮助小C、小S和小P来模拟实现上述过程，并对于每个操作输出操作结果。

输入格式

从标准输入读入数据。

第一行：3 个正整数 n, m, k。

接下来 k 行，每行若干个整数描述一个操作，格式如上所述。

输出格式

输出到标准输出。

输出共 k 行，对于每个操作输出一行。

对于每个写入操作，输出一个整数表示此次操作写入成功的最右位置；特别地如果该操作一个位置也没有写入成功，输出 -1。

对于每个删除、恢复操作，若该操作成功，输出一个字符串 `OK`，否则输出一个字符串 `FAIL`。

对于每个读取操作，输出两个整数表示此次查询的结果。

样例1输入

```
3 15 12
0 1 1 5 -1
0 2 10 13 2
0 1 4 14 6
1 1 2 8
3 1
3 3
3 14
2 1 3 5
0 3 7 8 -4
2 1 6 8
1 3 6 7
0 2 5 7 3
```

样例1输出

```
5
13
9
OK
1 -1
0 0
0 0
OK
8
FAIL
FAIL
-1
```

样例2

见disk_down.zip (/staticdata/disk_down.zip)下的 *disk2.in* 与 *disk2.ans*。

样例3

见disk_down.zip (/staticdata/disk_down.zip)下的 *disk3.in* 与 *disk3.ans*。

子任务

对于 25% 的数据，$n, k \leq 2000,\ m \leq 10000$；

对于另外 15% 的数据，没有删除、恢复操作；

对于另外 20% 的数据，没有恢复操作；

对于另外 15% 的数据，$n = 1$；

对于 100% 的数据，
$1 \leq n, k \leq 2 \times 10^5,\ 1 \leq m \leq 10^9,\ 1 \leq id \leq n,\ 1 \leq l \leq r \leq m,\ 1 \leq p \leq m,\ |x| \leq 10^9$ 。

语言及编译选项信息

#	名称	编译器	额外参数	代码长度限制（B）
0	g++ with std11	g++	-O2 -std=c++11 -DONLINE_JUDGE	65536
1	g++	g++	-O2 -DONLINE_JUDGE	65536
2	gcc with std11	gcc	-O2 -std=c11 -DONLINE_JUDGE	65536
3	gcc	gcc	-O2 -DONLINE_JUDGE	65536

#	名称	编译器	额外参数	代码长度限制（B）
4	java	javac		65536
5	python	python		65536
6	python3	python3		65536

递交历史

#	状态	时间
9949	**Waiting** (/#!/contest/13/detail/9949)	02:57:12 PM **有效递交**
9941	**Running on case 1** (/#!/contest/13/detail/9941)	02:56:52 PM
9814	**Wrong Answer** (/#!/contest/13/detail/9814)	02:54:09 PM
9744	**Runtime Error** (/#!/contest/13/detail/9744)	02:52:32 PM
9741	**Runtime Error** (/#!/contest/13/detail/9741)	02:52:24 PM
9692	**Runtime Error** (/#!/contest/13/detail/9692)	02:51:24 PM
9683	**Memory Limit Exceeded** (/#!/contest/13/detail/9683)	02:51:05 PM

#	状态	时间
9613	**Wrong Answer** (/#!/contest/13/detail/9613)	02:49:19 PM
9586	**Wrong Answer** (/#!/contest/13/detail/9586)	02:48:42 PM
9576	**Wrong Answer** (/#!/contest/13/detail/9576)	02:48:19 PM

1 2 3

递交答案 (剩余次数: 32)

语言和编译选项

g++ with std11

1

递交评测

文件请拖入编辑器中，或 上传文件