# 1861: [Zjoi2006]Book 书架

Time Limit: 4 Sec Memory Limit: 64 MB Submit: 1078 Solved: 626 [Submit][Status][Discuss]

#### **Description**

小T有一个很大的书柜。这个书柜的构造有些独特,即书柜里的书是从上至下堆放成一列。她用1到n的正整数给每本书都编了号。 小T在看书的时候,每次取出一本书,看完后放回书柜然后再拿下一本。由于这些书太有吸引力了,所以她看完后常常会忘记原来是放在书柜的什么位置。不过小T的记忆力是非常好的,所以每次放书的时候至少能够将那本书放在拿出来时的位置附近,比如说她拿的时候这本书上面有X本书,那么放回去时这本书上面就只可能有X-1、X或X+1本书。 当然也有特殊情况,比如在看书的时候突然电话响了或者有朋友来访。这时候粗心的小T会随手把书放在书柜里所有书的最上面或者最下面,然后转身离开。 久而久之,小T的书柜里的书的顺序就会越来越乱,找到特定的编号的书就变得越来越困难。于是她想请你帮她编写一个图书管理程序,处理她看书时的一些操作,以及回答她的两个提问: (1)编号为X的书在书柜的什么位置; (2)从上到下第i本书的编号是多少。

# Input

第一行有两个数n, m, 分别表示书的个数以及命令的条数; 第二行为n个正整数: 第i个数表示初始时从上至下第i个位置放置的书的编号; 第三行到m+2行,每行一条命令。命令有5种形式: 1. Top S——表示把编号为S的书房在最上面。 2. Bottom S——表示把编号为S的书房在最下面。 3. Insert S T——T∈{-1,0,1}, 若编号为S的书上面有X本书,则这条命令表示把这本书放回去后它的上面有X+T本书; 4. Ask S——询问编号为S的书的上面目前有多少本书。 5. Query S——询问从上面数起的第S本书的编号。

#### **Output**

对于每一条Ask或Query语句你应该输出一行,一个数,代表询问的答案。

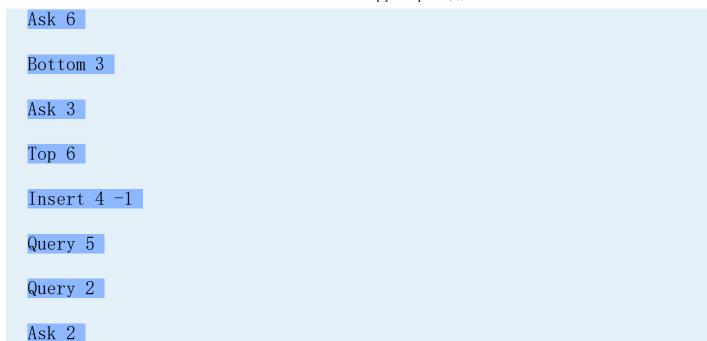
#### **Sample Input**

10 10

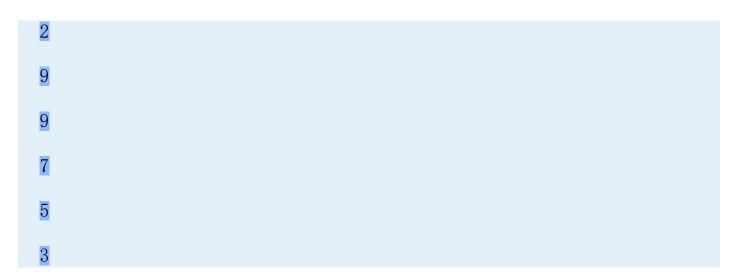
1 3 2 7 5 8 10 4 9 6

Query 3

Top 5



# **Sample Output**



# **HINT**

数据范围

100%的数据,n,m < = 80000

# **Source**

Day2

[Submit][Status][Discuss]