

Given an array of integers  $A$ , you are to process  $C$  commands. Each command can be one of three types:

- ‘sum  $i\ j$ ’: output the summation of all integers in the array with indices between  $i$  to  $j$ , inclusive.
- ‘insert  $i$  <description of  $B$ >’: insert a new array  $B$  into array  $A$  at the position immediately preceding  $A[i]$ . The exact format of the description of  $B$  in the input is provided below.
- ‘delete  $i\ j$ ’: delete the part of the given array between  $i$  and  $j$ , inclusive.

All indices in the input will be 0-based. For each scenario, the starting array is always an empty array of size 0, since the first command will always be an insert command at position 0 (meaning start of array).

Every insert command will describe an array of size  $n$  with  $r + 5$  space-separated integers all on the same line of input. The first five of these integers are  $n$ ,  $r$ ,  $m$ ,  $a$  and  $c$ , in that order. The rest of the line will contain  $r$  more integers:  $b[0]$ ,  $b[1]$ ,  $\dots$ ,  $b[r - 1]$ . These are the first  $r$  elements of the array  $B$ . The rest can be obtained by using the following scheme:

for each  $i$  from  $r$  to  $n - 1$ , inclusive:

$$b[i] = (b[i - r] * a + c) \bmod m$$

$$c = ((b[i - r] * a + c) \operatorname{div} m) \bmod m$$

where “ $a \bmod m$ ” is the remainder after dividing  $a$  by  $m$ , while “ $a \operatorname{div} m$ ” is the quotient.

## Input

The input consists of upto 25 scenarios. Each scenario starts with an integer  $C$ , followed by  $C$  lines of input, each specifying a command. Each command will be formatted in one of the three ways described above. For each ‘sum’ and ‘delete’ command, the indices  $i$  and  $j$  will satisfy the condition  $0 \leq i \leq j < N$ , where  $N$  is the array size before that command is executed. For each ‘insert’ command, the index  $i$  will be between 0 and  $N$  inclusive, where position  $N$  refers to the end of the array. Every other integer in an insert command will be positive and small enough to fit in a signed 32-bit integer;  $r$  will be  $\leq \min(n, 100)$ . You also may assume that  $C \leq 500$ , and that the sum of the size (or the ‘ $n$ ’ value) of all inserts together will not exceed 2000000. The input ends with a ‘0’ on a line by itself, which should not be processed.

## Output

For each scenario, the output should begin with a single line of the form ‘Scenario  $k$ :’, where  $k$  is the scenario serial number, starting with 1 for the 1st scenario, 2 for the 2nd scenario, and so on. This line should be followed by one line of output for each ‘sum’ command, with just the sum on the line by itself. The output for every scenario should be followed by a blank line.

## Sample Input

```
11
insert 0 10 10 100 100 100 1 2 3 4 5 6 7 8 9 10
sum 0 9
sum 3 8
insert 3 10 10 100 100 100 10 20 30 40 50 60 70 80 90 100
sum 2 9
sum 2 15
sum 0 19
delete 2 15
sum 0 5
sum 1 5
sum 0 3
2
insert 0 50 5 1000 91 7 1 2 3 4 5
sum 0 49
0
```

## Sample Output

```
Scenario 1:
55
39
283
568
605
37
36
18

Scenario 2:
21356
```