5530. Fancy Sequence

| My Submissions (/contest/biweekly-contest-37/problems/fancy-sequence/submissions/) | Back to Contest (/contest/biweekly-contest-37/) |

Write an API that generates fancy sequences using the `append` , `addAll` , and `multAll` operations.

Implement the `Fancy` class:

User Accepted:	0
User Tried:	0
Total Accepted:	0
Total Submissions:	0
Difficulty:	Hard

- `Fancy()` Initializes the object with an empty sequence.
- `void append(val)` Appends an integer `val` to the end of the sequence.
- `void addAll(inc)` Increments all existing values in the sequence by an integer `inc` .
- `void multAll(m)` Multiplies all existing values in the sequence by an integer `m` .
- `int getIndex(idx)` Gets the current value at index `idx` (0-indexed) of the sequence **modulo** $10^9 + 7$. If the index is greater or equal than the length of the sequence, return -1 .

Example 1:

```
Input
["Fancy", "append", "addAll", "append", "multAll", "getIndex", "addAll", "append", "multAll", "getIndex", "getIndex", "get
[[], [2], [3], [7], [2], [0], [3], [10], [2], [0], [1], [2]]
Output
[null, null, null, null, null, 10, null, null, null, 26, 34, 20]

Explanation
Fancy fancy = new Fancy();
fancy.append(2);   // fancy sequence: [2]
fancy.addAll(3);   // fancy sequence: [2+3] -> [5]
fancy.append(7);   // fancy sequence: [5, 7]
fancy.multAll(2);  // fancy sequence: [5*2, 7*2] -> [10, 14]
fancy.getIndex(0); // return 10
fancy.addAll(3);   // fancy sequence: [10+3, 14+3] -> [13, 17]
fancy.append(10);  // fancy sequence: [13, 17, 10]
fancy.multAll(2);  // fancy sequence: [13*2, 17*2, 10*2] -> [26, 34, 20]
fancy.getIndex(0); // return 26
fancy.getIndex(1); // return 34
fancy.getIndex(2); // return 20
```

Constraints:

- `1 <= val, inc, m <= 100`
- `0 <= idx <= ` 10^5
- At most 10^5 calls total will be made to `append` , `addAll` , `multAll` , and `getIndex` .

| JavaScript ▾ | | 〈⁄〉 ⟳ ⚙ |

```javascript
1
2 ▾ var Fancy = function() {
3
4   };
5
6 ▾ /**
7    * @param {number} val
8    * @return {void}
9    */
10 ▾ Fancy.prototype.append = function(val) {
11
12   };
13
14 ▾ /**
15    * @param {number} inc
16    * @return {void}
17    */
18 ▾ Fancy.prototype.addAll = function(inc) {
```

```
19
20   };
21
22 ▼ /**
23    * @param {number} m
24    * @return {void}
25    */
26 ▼ Fancy.prototype.multAll = function(m) {
27
28   };
29
30 ▼ /**
31    * @param {number} idx
32    * @return {number}
33    */
34 ▼ Fancy.prototype.getIndex = function(idx) {
35
36   };
37
38 ▼ /**
39    * Your Fancy object will be instantiated and called as such:
40    * var obj = new Fancy()
41    * obj.append(val)
42    * obj.addAll(inc)
43    * obj.multAll(m)
44    * var param_4 = obj.getIndex(idx)
45    */
```

☐ **Custom Testcase**   (Use Example Testcases)

(▶ Run)   (☁ Submit)