Product of the Maximum and Minimum in a Dataset

Starting with an empty set of integers named *elements,* perform the following query operations:

* The command *push x* inserts the value of *x* into *elements*.
* The command *pop x* removes the value of *x* from *elements*.

The integers in *elements* need to be ordered in such a way that after each operation is performed, the product of the maximum and minimum values in the set can be easily calculated.

**Function Description**

Complete the function *maxMin* in the editor below.

maxMin has the following parameter(s):

*string operations[n]:*  an array of operations strings

*int x[n]: an* array of *x* where *x[i]* goes with *operations[i]*.

Returns:

*int[n]*: an array of long integers that denote the product of the maximum and minimum of *elements* after each query

**Constraints**

* *1 ≤ n ≤ 105*
* *1 ≤ x[i] ≤ 109*
* It is guaranteed that each *operations[i]* is either *push* or *pop*.
* It is guaranteed that any value popped will exist in the array.

**Sample Case 0**

**Sample Input**

STDIN     Function   
-----     -----   
4      →  operations[] size n = 4   
push   →  operations = [push, push, push, pop]   
push   
push   
pop   
4      →  x[] size n = 4   
1      →  x = [1, 2, 3, 1]   
2   
3   
1 

**Sample Output**

1   
2   
3   
6 

**Explanation**

Visualize *elements* as an empty multiset, *elements = {}*, and refer to the return array as *products*. The sequence of operations occurs as follows:

1. *push 1* → *elements = {1}*, so the *minimum = 1* and the *maximum = 1*. Then store the product as *products0 = 1 × 1 = 1*.
2. *push 2* → *elements = {1, 2}*, so the *minimum = 1* and the *maximum = 2*. Then store the product as *products1 = 1 × 2 = 2*.
3. *push 3* → *elements = {1, 2, 3}*, so the *minimum = 1* and the *maximum = 3*. Then store the product as *products2 = 1 × 3 = 3*.
4. *pop 1* → *elements = {2, 3}*, so the *minimum = 2* and the *maximum = 3*. Then store the product as *products3 = 2 × 3 = 6*.

Return *products = [1, 2, 3, 6]*

**Sample Case 1**

**Sample Input**

STDIN     Function   
-----     -----   
2      →  operations[] size n = 2   
push   →  operations = [push, push]   
push   
2      →  x[] size n = 2   
3      →  x = [3, 2]   
2 

**Sample Output**

9   
6

**Explanation**

Visualize *elements* as an empty multiset, *elements = {}*, and refer to the return array as *products*. The sequence of operations occurs as follows:

1. *push 3* → *elements = {3}*, so the *minimum = 3* and the *maximum = 3*. Then store the product as *products0 = 3 × 3 = 9*.
2. *push 2* → *elements = {2, 3}*, so the *minimum = 2* and the *maximum = 3*. Then store the product as *products1 = 2 × 3 = 6*.

Return *products = [9, 6]*