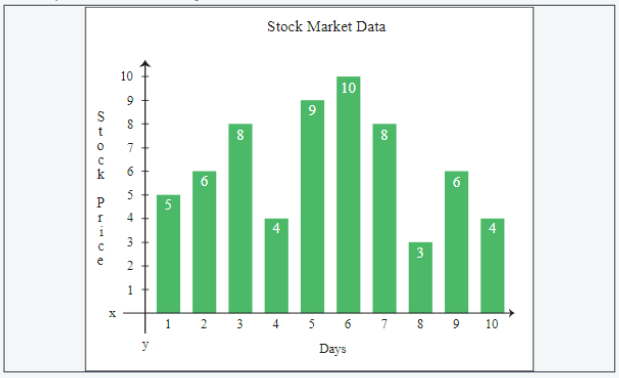
5. Stock market prediction

In this prediction game, the first player gives the second player some stock market data for some consecutive days. The data contains a company's stock price on each day. The rules for the game are:

- Player *1* will tell player *2* a day number
- player *2* has to find the nearest day on which stock price is smaller than the given day
- If there are two results, then player *2* finds the day number which is smaller
- if no such day exists, then the answer is *-1*.

For example, the image below shows the stock market data for *10* consecutive days. The horizontal axis represents the day number, starting at 1, and the vertical axis represents the stock price on that day.



Example
n = 10
stockData = [5, 6, 8, 4, 9, 10, 8, 3, 6, 4]
queries = [6, 5, 4]

On day 6, the stock price is 10. Both 9 and 8 are lower prices one day away. Choose 9 (day 5) because it is before day 6.
On day 5, the stock price is 9. 4 is the closest lower price on day 4.
On day 4, the stock price is 4. The only lower price is on day 8.
The return array is [5, 4, 8].

Function Description

Complete the *predictAnswer* function in the editor below.

predictAnswer has *2* parameters:
int stockData[n]: the value of each *stockData[i]* is the stock price on the $i+1^{th}$ day (where $0 \le i < n$).
int queries[q]: the value of each element *queries[j]* is the day number given in the query (where $0 \le j < q$).

Return
int[q]: the value at each index *i* is the answer to *queries[i]*

Constraints

- $1 \le n \le 10^5$
- $1 \le stockData[i] \le 10^9$
- $1 \le q \le 10^5$
- $1 \le queries[j] \le 10^9$

▼ Input Format For Custom Testing

Locked stub code in the editor reads the following input from stdin and passes it to the function.

The first line contains an integer, *n*, denoting the number of elements in *stockData*.
Each line i^{th} of the *n* subsequent lines contains an integer, *stockData[i]*, the stock price on the $i+1^{th}$ day.
Next line contains an integer, *q*, the number of elements in *queries*.
Each line j^{th} of the *q* subsequent lines contains an integer, *queries[j]*, the day number of the j^{th} query.

▼ Sample Case 0

Sample Input 0

```
STDIN       Function
-----       --------
10    →     stockData[] size n = 10
5     →     stockData = [5, 6, 8, 4, 9, 10, 8, 3, 6, 4]
6
8
4
9
10
8
3
6
4
3     →     queries[] size q = 3
3     →     queries = [3, 1, 8]
1
8
```

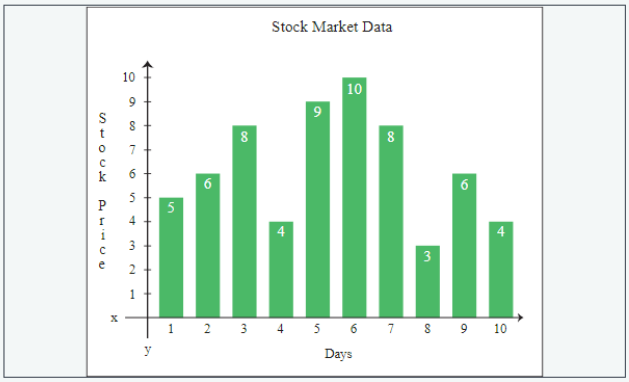
Sample Output 0

```
2
4
-1
```

Sample Output 0

```
2
4
-1
```

Explanation 0



- If the day number is *3*, both days 2 and 4 are smaller. Choose the earlier day, day 2.
- If the day number is *1*, day 4 is the closest day with a smaller price.
- If the day number is *8*, there is no day where the price is less than 3. The answer is -1.
- The return array is [2, 4, -1]