Signal Classification

A radio center received some signals and need to classify them according to frequency.

There are n standard frequencies known to the center. They have identified q different signals which are to be classified. Given the standard signal frequencies *freq_standard* and frequencies of signals to be classified *freq_signal*, can you help the radio center identify them?

A signal X belongs to a standard signal Y if the frequency of X is closer to that of Y than to any other frequency. If it is equidistant from two known frequencies, then the signal with higher frequency is chosen.

Consider, for example, *freq_standard* = [2, 3, 1, 4, 8] and *freq_classify* = [1, 5, 6]. Frequencies 1 and 5 belong to standard frequencies 1(index=3) and 4(index=4) respectively. Since 6 is equidistant from two standard frequencies, 4 and 8, choose the higher frequency, 8(index=5). The corresponding classifications are [3, 4, 5].

Function Description

Complete the function *classifySignals* in the editor below. The function must return an integer array denoting the classifications of each frequency.

classifySignals has two parameters -

freq_standard: an integer array

freq_signals: an integer array

Input Format

The first line of input contains 2 space-separated integers: n, q - the number of strings and the number of queries.

The second line contains n space-separated integers, the array *freq_standard*.

The next line contains q space-separated integers, the array *freq_signals*.

Constraints

- $n \leq 10^5$
- $q \leq 10^5$
- |freq_standard$_i$| $\leq 10^9$
- |freq_signals$_i$| $\leq 10^9$

Output Format

Print q lines: each line should contain an integer representing the index of the standard frequency corresponding to the given signal. There is a code stub to handle I/O if you choose to use it.

Sample Input 0

```
5 5
7 1 12 9 15
2 9 2000 13 4
```

Sample Output 0

```
2
4
5
3
1
```

Explanation 0

2 is closest to 1(index=2)

9 is closest to 9(index=4)

2000 is closest to 15(index=5)

13 is closest to 12(index=3)

4 is closest to both 7 and 1. Since 7 is greater, we consider 7(index=1).

Sample Input 1

```
2 10
1 2
1 2 3 4 5 6 7 8 9 10
```

Sample Output 1

```
1
2
2
2
2
2
2
2
2
2
```

Explanation 1

1 is closest to 1.

All other numbers are closer to 2.