Minimum Multiplications to reach End

Given **start**, **end** and an array **arr** of **n** numbers. At each step, **start** is multiplied with any number in the array and then mod operation with **100000** is done to get the new start.

Your task is to find the minimum steps in which **end** can be achieved starting from **start**. If it is not possible to reach **end**, then return **-1**.

Example 1:

```
Input:
arr[] = {2, 5, 7}
start = 3, end = 30
Output:
2
Explanation:
Step 1: 3*2 = 6 % 100000 = 6
Step 2: 6*5 = 30 % 100000 = 30
```

Example 2:

```
Input:
arr[] = {3, 4, 65}
start = 7, end = 66175
Output:
4
Explanation:
Step 1: 7*3 = 21 % 100000 = 21
Step 2: 21*3 = 63 % 100000 = 63
Step 3: 63*65 = 4095 % 100000 = 4095
Step 4: 4095*65 = 266175 % 100000 = 66175
```

Your Task:

You don't need to print or input anything. Complete the function **minimumMultiplications()** which takes an integer array **arr**, an integer **start** and an integer **end** as the input parameters and returns an integer, denoting the minumum steps to reach in which **end** can be achieved starting from **start**.

Expected Time Complexity: $O(10^5)$

Expected Space Complexity: $O(10^5)$

Constraints:

- $1 <= n$ and $n <= 10^4$
- $1 <= arr[i]$ and $arr[i] <= 10^4$
- $1 <= start, end < 10^5$