Uber is building a Fare Estimator that can tell you how much your ride will cost before you request it. It works by passing approximated ride distance and ride time through this formula:

`(Cost per minute) * (ride time) + (Cost per mile) * (ride distance)`
where `Cost per minute` and `Cost per mile` depend on the car type.
You are one of the engineers building the Fare Estimator, so knowing cost per minute and cost per mile for each car type, as well as ride distance and ride time, return the fare estimates for all car types.

Example

For
`ride_time = 30`,
`ride_distance = 7`,
`cost_per_minute = [0.2, 0.35, 0.4, 0.45]` and
`cost_per_mile = [1.1, 1.8, 2.3, 3.5]`, the output should be
`fareEstimator(ride_time, ride_distance, cost_per_minute, cost_per_mile) = [13.7, 23.1, 28.1, 38]`.
Since:

```
30 * 0.2 + 7 * 1.1 = 6 + 7.7 = 13.7

30 * 0.35 + 7 * 1.8 = 10.5 + 12.6 = 23.1

30 * 0.4 + 7 * 2.3 = 12 + 16.1 = 28.1

30 * 0.45 + 7 * 3.5 = 13.5 + 24.5 = 38
```

Input/Output

- **[time limit] 4000ms (py)**
- **[input] integer ride_time**

  A positive integer, ride duration in minutes.

  Constraints:
  `10 ≤ ride_time ≤ 50`.
- **[input] integer ride_distance**

  A positive integer, ride distance in miles.

  Constraints:
  `5 ≤ ride_distance ≤ 20`.
- **[input] array.float cost_per_minute**

  `cost_per_minute[i]` is a positive number denoting cost per minute for the i^{th} car type. There are at least 4 car types in every city where Uber operates.
  Constraints:
  `4 ≤ cost_per_minute.length ≤ 10`,
  `0.1 ≤ cost_per_minute[i] ≤ 350.0`.

- **[input] array.float cost_per_mile**

  `cost_per_mile[i]` is a positive number denoting cost per mile for the i^{th} car type. It is guaranteed that `cost_per_minute` and `cost_per_mile` have the same number of elements.
  Constraints:
  `cost_per_mile.length = cost_per_minute.length`,
  `0.5 ≤ cost_per_mile[i] ≤ 7.0`.

- **[output] array.float**

  An array of estimated fares for each car type.