

Question Activated

Problem Statement

You are selling a number of boxes of berries to a distributor. You are paid per berry, but the distributor only knows how many boxes you are shipping - not how many berries are in each box. To get around this, the distributor employs inspectors to estimate the total number of berries. Each inspector will take a sample of the boxes, count the berries in each of the sample boxes, and use this to extrapolate the total number of berries to pay for. It is up to you to decide how many berries to put in each individual box. The total number of berries to package is given in **berries**, and all berries must be used. You can put between 1 and 9 berries, inclusive, in each box.

Each inspector n starts at the box numbered **first**[n] and then proceeds to box **first**[n]+**period**[n] and continues inspecting boxes until reaching the end of the boxes (boxes begin numbering at zero). If an inspector were to see a total of 30 berries in the 20 boxes looked at, and if there were 25 boxes altogether, that inspector would give an estimate of $30 * 25 / 20 = 37.5$ berries for the entire shipment. If an inspector does not open any boxes, that inspector will estimate that there are zero berries in the shipment. More than one inspector may look at the same box. You will be paid based on the average estimate of the inspectors.

Since you know which boxes each inspector will look at, it is in your best interest to put more berries in those boxes and to choose a number of boxes that will make your shipment seem as large as possible. Assuming that you do so, what is the maximum number of berries you will be paid for?

Definition

Class: BerryPacker
Method: bestPacking
Parameters: int[], int[], int
Returns: double
Method signature: double bestPacking(int[] first, int[] period, int berries)
(be sure your method is public)

Constraints

- **first** and **period** will each contain between 1 and 5 elements, inclusive.
- **first** and **period** will contain the same number of elements.
- Each element of **first** will be between 0 and 100,000 inclusive.
- Each element of **period** will be between 1 and 100,000 inclusive.
- **berries** will be between 1 and 100,000 inclusive.

Examples

0)

```
{2}
{500}
6
Returns: 12.0
```

There's only one inspector, and he's only going to see one box - box 2 (you don't have nearly enough berries to think about box 502). One way to maximize your payment is to use 4 boxes, put 3 berries in box 2 and 1 berry in each of the other 3 boxes. Inspector sees 3 berries in one box and assumes there are 12 in shipment.

1)

```
{0,1}
{2,2}
7
Returns: 9.0
```

Between the two inspectors, each box is going to get looked at once (inspector 0 looks at even boxes, inspector 1 at odd boxes). Your best bet is to put 5 berries in box 1 and 1 berry in boxes 0 and 2. That way inspector 1 sees 5 berries and assumes there's 15. Inspector 0 sees 2 berries in 2 boxes, and estimates a shipment of 3. $(15+3)/2=9$

2)

```
{2,5,9,25}
{1,3,11,7}
100
Returns: 251.50649350649354
```