Problem Statement (SellingGold)

Jack has a store where he is selling gold. He has several boxes of gold and the i-th box contains **gold[i]** grams of gold and costs **price[i]** dollars.

Two customers entered the shop and want to buy all the gold. Jack knows that the total number of boxes is even, so he decided to give each customer half of the boxes. He knows that after buying gold, each customer will calculate the average price that he has paid for 1 gram of gold and announce it. Jack wants the sum of the two announced numbers to be as small as possible.

Given int[]'s **gold** and **price**, return the minimal possible sum of two described averages.

Definition

Class:	SellingGold
Method:	minimalSum
Parameters:	int[], int[]
Returns:	double
Method signature:	double minimalSum(int[] gold, int[] price)
(be sure your method is public)	

Notes

- Your return value must have an absolute or relative error less than 1e-9.

Constraints

- **gold** will contain between 2 and 50 elements, inclusive.
- **gold** will contain an even number of elements.
- **gold** and **price** will contain the same number of elements.
- Each element of **gold** will be between 1 and 500, inclusive.
- Each element of **price** will be between 1 and 500, inclusive.

Examples

0)
```
{2,4}

{3,4}

Returns: 2.5
```
The average price for one of the customers will be 3/2=1.5, and for the other one it will be 4/4=1. The answer is 1.5+1=2.5.

1)
```
{1,1,2,2}

{1,1,1,1}

Returns: 1.3333333333333333
```
There are two different ways to divide the boxes among the customers. In the first case one of the customers gets the first two boxes and the other customer gets two remaining boxes. In this case sum of the averages is 2/2+2/4=1.5. In the second case each customer is sold 3 grams for 2 dollars. Therefore we get 2/3+2/3=1.33(3) which is optimal.

2)
```
{2,1,1,1}

{300,300,300,300}

Returns: 500.0
```
Remember that each customer must get exactly half of the boxes.

3)
```
{500,500,500,500,500,500}

{1,2,4,4,2,2}

Returns: 0.01
```
It doesn't matter how the boxes are divided.

4)
```
{1,2,3,4,5,6,7,8}

{2,3,4,5,6,7,8,10}

Returns: 2.498452012383901
```