You are planning a drive along the Yellow Brick Road from Munchkinland to Emerald City. There are $n + 1$ hotels along the route, numbered 0 through n. Hotel 0 is in Munchkinland and hotel n is in Emerald City.

The distance in miles from Munchkinland to hotel i is distance[i], where $0 =$ distance[0] $<$ distance[1] $< \ldots . <$ distance[n]. The first hotel is in Munchkinland and is (of course) zero miles from Munchkinland, and the final hotel is in Emerald City.

You start in Munchkinland on the morning of day 1, having stayed in the Munchkinland hotel the night before. At the end of each day you must stay in one of the hotels; at the end of the last day you must stay in the Emerald City hotel; you may never turn around and drive back toward Munchkinland.

The Good Witch of the North is paying for your trip. She doesn't want you to drive too far each day (that isn't safe) or too little each day (that's too expensive). She wants you to drive approximately 400 miles each day, so she makes the following deal. Each day you will drive some distance x to reach a hotel. You will owe her a penalty of $(400 - x)^2$ cents for that day. At the end of the trip you will owe her the sum of all the penalties you have amassed.

You, of course, want to minimize the total penalty. For example, suppose

```
distance = [0, 350, 450, 825].
```

The optimal strategy is to drive to hotel 2 on the first day (penalty 2500) and then to hotel 3 the second day (penalty 625), for a total penalty of 3125 cents.

On the other hand, suppose

```
distance = [0, 350, 450, 700].
```

The optimal strategy is to drive to hotel 1 on the first day (penalty 2500) and then hotel 3 the second day (penalty 2500), for a total penalty of 5000 cents.

Given an array of distances, you are to determine the minimum possible penalty that can accumulate when driving from Munchkinland to Emerald City.

Input

The first line contains n, where $1 \leq n \leq 1000$. The next $n + 1$ lines give the elements of the hotel distance array described above. The first element is 0; each successive element is larger than its predecessor; the last element is less than 30000.

Output

Produce a single line of input that contains the minimum penalty for the trip.