UCC Coding Competition '20 P4 - Bubble Tea

Time limit: 1.5s
Python: 3.0s
Memory limit: 128M

In a group of N friends (including yourself), you travel to the nearest bubble tea store to buy some bubble tea. Everyone picks out their desired bubble tea and lines up to order them. The cashier tells you that there is a special promotion going on: for a purchase of 2 bubble teas, the cheaper one is discounted at 25% off, and for a purchase of 3 bubble teas, the cheapest one of the three is discounted at 50% off.

You and your friends are already in line, and it is **no longer possible to change the order of this line**. However, you can still change how the orders are placed – the 2 or 3 people at the front of the line can group together and place their order together to take advantage of one of the offers (but it is still possible for the person at the front to pay the full price by ordering for themselves only).

The way that you order could change the total amount the entire group pays. For example, if you are in a group of 3 and the prices for the 3 people's bubble tea is 100, 100 and 4 (and they are lined up in that order), you can pay in many ways:

- Everyone orders themselves: $100 + 100 + 4 = 204$ total cost
- First two people order together: $(100 + 100 - 0.25 \times 100) + 4 = 179$ total cost
- Last two people order together: $100 + (100 + 4 - 0.25 \times 4) = 203$ total cost
- Three people order together: $(100 + 100 + 4 - 0.50 \times 4) = 202$ total cost

In this case, the optimal price for all the bubble tea combined is 179, which occurs when the first two people place their orders together and the third person orders for themselves.

You and your friends decide to order the bubble teas in a way such that the most money is saved. Please help them find the minimum amount of money needed for the **whole group to buy all the bubble tea optimally by strategically grouping themselves at the checkout**. For your convenience, **all the prices of the bubble tea will be multiples of 4**.

Input Specification

The first line will contain the integer N $(1 \leq N \leq 10^6)$, the number of friends lining up to buy bubble tea.

The next N lines will consist of one integer each: the price of each friend's bubble tea, in the order that they lined up to check out. As you and your friends are rather modest, no bubble tea will cost more than 1000.

Due to large input sizes, Python users should try the PyPy interpreter to avoid timing out.

Output Specification

Output one integer, the price the entire group pays to buy all the bubble tea, assuming they take advantage of the discounts optimally to minimize this total cost.

Constraints

- For 2 out of 10 available marks, $N \leq 12$.
- For another 4 out of 10 available marks, $N \leq 5000$.
- For the remaining 4 marks, there are no further constraints.

Sample Input 1

```
3
100
100
4
```

Sample Output 1

```
179
```

Explanation for Sample Output 1

This was discussed in the problem statement.

Sample Input 2

```
4
40
1000
1000
1000
```

Sample Output 2

```
2540
```

Explanation for Sample Output 2

We consider all options:

- All 4 persons pay for themselves (no discounts): 3040
- Persons 1 and 2 go for the 25% discount, persons 3 and 4 go for the 25% discount: 2780
- Persons 1 and 2 go for the 25% discount, while persons 3 and 4 pay individually: 3030

- Persons 1 and 2 pay individually, while persons 3 and 4 go for the 25% discount: 2790
- Persons 2 and 3 go for the 25% discount, while persons 1 and 4 pay individually: 2790
- Persons 1-3 go for the 50% discount: 3020
- Persons 2-4 go for the 50% discount: 2540

The minimum cost is 2540.