Save-It

Input File: *savein.txt*
Output File: *saveout.txt*

Time Limit: 1 second

The United Bakers of Australia (UBA) is once again in hot water. Profits have been crumbling recently due to a new tax on an invisible poisonous ingredient underpinning cake manufacturing. Rather than adapt their recipes to use healthier alternatives, the UBA has employed analysts from PwC (People with Cupcakes) to find a way to cut costs.

Upon inspection of UBA's purchasing practices, the analysts found that when a group of ingredients is ordered, the cost of the group is rounded to the nearest 5 cents. That is, a group's cost is the multiple of 5 that has the smallest difference with the sum of its ingredient costs.

For instance, two items costing 32c and 47c can be bought in separate groups for 30c and 45c, or 75c in total, whereas purchasing them in one group together would cost 80c (rounded up from 79c).

Your task is to put the required ingredients into groups for purchasing in order to minimise their total cost.

Input

Your program should read from the file `savein.txt`. The file will describe a single list of ingredients.

- The first line of input will contain one integer N: the number of ingredients.
- The next N lines each contain one integer c_i: the cost of the ith ingredient in cents.

Output

Your program should write to the file `saveout.txt`. Your output file should contain one line with one integer: the minimum total cost for the ingredients.

Sample Input 1

```
4
31
103
14
21
```

Sample Output 1

```
165
```

Explanation

Buy the first and fourth ingredients together for 31c + 21c = 52c, which rounds down to 50c. Then buy the second and third ingredients together for 103c + 14c = 117c, which rounds down to 115c. This totals to 50c + 115c = 165c.

Sample Input 2

```
3
18
18
20
```

Sample Output 2

```
55
```

Explanation

One way to obtain the minimal cost is to buy all items in the same group for 18c + 18c + 20c = 56c, which rounds to 55c.

Constraints

To evaluate your solution, the judges will run your program against several different input files. All of these files will adhere to the following bounds:

- $1 \leq N \leq 100\,000$ (the number of ingredients)
- $1 \leq c_i \leq 1\,000$ (the price of the ith ingredient in cents)

As some of the test cases will be quite large, you may need to think about how well your solution scales for larger input values. However, not all the cases will be large. In particular:

- For 20% of the marks, $N \leq 3$.
- For an additional 20% of the marks, $N \leq 20$.
- For an additional 30% of the marks, $N \leq 3\,000$.
- For the remaining 30% of the marks, no special constraints apply.