Cupid's job is getting harder, so he is adopting new technologies to help him with his difficult task of matching people into happy couples. He appointed the best programmers in his staff to a new project called Advanced Couples Matching (ACM). For this project, the programmers need to produce an algorithm that takes a set of an even number of $N$ lonely persons and matches them into $N/2$ couples, such that each person is in exactly one couple.

Sadly, the data available about each person is limited. In this modern world, using gender, ethnicity, age or nationality as criteria to form couples is not a sensible option, so the programmers can only use data about the internet connection of each candidate. They decided to focus this stage on time zones. People living in closer time zones are more likely to find time to interact with each other. Thus, the programmers decided to create couples so as to minimize the *total time difference*.

Each time zone is identified by an integer between -11 and 12, inclusive, representing its difference in hours from a particular time zone called Coordinated Universal Time (or UTC). The time difference of two people living in time zones represented by integers $i$ and $j$ is the minimum between $|i - j|$ and $24 - |i - j|$. Given a partition of a set of an even number $N$ of candidates into $N/2$ couples, its total time difference is the sum of the time difference of each couple.

You are asked to write a program that receives as input the time zones of a set of $N$ candidates. The output of the program must be the minimum total time difference among all possible partitions of the set into couples.

## Input

The input contains several test cases; each test case is formatted as follows. The first line contains an even integer $N$ ($2 \leq N \leq 1000$) representing the number of candidates to be coupled. The second line contains $N$ integers $T_1, T_2, \ldots, T_N$ ($-11 \leq T_i \leq 12$ for $i = 1, 2, \ldots, N$) indicating the time zones of the candidates.

## Output

For each test case in the input, output a line with an integer representing the minimum total time difference among all possible partitions of the set of candidates into couples.

## Sample Input

```
6
-3 -10 -5 11 4 4
2
-6 6
8
0 0 0 0 0 0 0 0
```

## Sample Output

```
5
12
0
```