Problem ELEVCONSTR: Elevator Construction

You might know the "*Blaues Hochhaus*" (blue skyscraper) where most of the computer scientists of the FAU are working. There are three elevators in the building and most of the time at least one of the elevators is out of order and has to be maintained. A lesson learned from the construction of the new canteen, which was not finished in almost two years and still is delayed, the computer scientists do not risk the construction of completely new elevators.

After some research, the scientists discovered that the software is the main problem and far too complex. Thus, they decided to simplify the software part as follows: if a group of n persons arrives at the three elevators (in floor 0), every person specifies his/her target floor. Then, the group is split in four (possibly empty) subgroups, one for each elevator, and one that does not take the elevator at all. Every elevator goes either up or down (if possible) and stops in exactly one floor. Afterwards the elevators return to the default position (floor 0).
If the elevator does not stop at your floor, you have to walk up/or down the difference. Your job is to compute which elevator should stop at which floor while minimizing the total walked floors.

Input

The first line denotes the number of test cases $1 \leq t \leq 100$. Each test case consists of exactly two lines. The first line of each test case contains the number of persons n in a group ($1 \leq n \leq 50$). The second line contains n integers, where the ith integer specifies the target floor of the ith person. The target floor is an integer between -1 and 12 inclusively.

Output

For each test case, print one line containing four integers. The first integer gives the total walked floors for all persons in that group. The other three integers specify the floor number where the elevator stops (from left to right). The first two elevators can stop at floors 0 to 12 excluding floor 3. The third elevator can stop at floors -1 to 12. If there are multiple solutions that minimize the total walked floors, any will be accepted.

Sample Input 1

```
3
5
3 3 7 10 12
4
-1 3 3 12
4
-1 3 7 12
```

Sample Output 1

```
2 12 7 3
1 8 12 3
1 7 12 3
```