# 12904 Load Balancing

The infamous University of Kala Jadu (UKJ) have been operating underground for the last fourteen centuries training very select few students the dangerous art of black magic. However, with the recent trend of going digital, they too wanted to try out a bit of public exposure by enrolling students in their new distance education program.

Within the first three semesters they have got 6,789 students enrolled. The forkaclone spell allows them to clone their teachers to train at most 10,000 students in a running semester. But they do not have the server capacity to handle 10,000 students to register for their courses right at the beginning of the semester during their 4-day registration period. Sadly, their art of black magic (or kala jadu, as they say), only works on humans, it cannot be extended to their web server running on a Pentium IV machine.

UKJ server administrators realized that if they could split the load on the server and balance it somehow, then they can still handle 10,000 students per semester. Their idea is to divide the students into roughly 4 equal groups A, B, C and D. Each group would then be given one day to register; no other group can register on that same day — they will get their turn. They wanted to use total number of credits completed by a student as the deciding factor to assign students to the 4 different groups. As the students who would register can have completed any integer number of credits between 0 to 160, one easy group assignment would be:

0 - 40 credits completed: group A
41 - 80 credits completed: group B
81 - 120 credits completed: group C
121 - 160 credits completed: group D

A bit of analysis of the number of students that may fall in these groups revealed that the number of students in each group vary greatly. So this particular idea of splitting students into 4 groups to balance the server load does not quite work out.

UKJ seeks your help in finding the credit boundaries that can create an optimal distribution of students so that each group roughly have the same number of students. You'd suggest three integers $a$, $b$ and $c$ to distribute the students as follows:

0 - $a$ credits completed: group A
$a + 1$ - $b$ credits completed: group B
$b + 1$ - $c$ credits completed: group C
$c + 1$ - 160 credits completed: group D

If the total number of students is $N$, the best possible scenario would place $N/4$ students in each group. You need to minimize the sum of difference, $d$ between $N/4$ and the number of student you place in each group. For example, given $N = 8$ students to distribute, if you divide them into a group of 3, 0, 3, 2 students then the difference with $N/4$ for the groups would be 1, 2, 1, 0 respectively. This results in $1 + 2 + 1 + 0 = 4$ as sum of differences. This is what you'd have to minimize. Note that, $N/4$ can be a floating point number.

## Input

The input description for the problem starts with $T$ $(1 < T \leq 100)$ — the number of test cases, then $T$ test cases follow. The first line of each case starts with the number of students $N$ $(0 < N \leq 10000)$. The next $N$ lines contains the number of credits (always integer), $C_i$ $(0 \leq C_i \leq 160)$ the $i$-th student have completed prior to this registration.

## Output

Output for each test case will start with the test case label (starting with 1, and formatted as shown in sample output.) The label will be followed by three integers, $a$, $b$ and $c$ ($0 \le a < b < c < 160$) denoting the group boundaries as described in the problem. If there are multiple such boundaries possible with the same $d$ value, then pick the solution that has the smallest $a$ value. If there is a tie, then pick the one with the smallest $b$ value. If even that fails to break the tie, then pick the solution with the smallest $c$ value.

## Sample Input

```
2
8
0
40
41
80
85
120
150
155
9
0
40
41
80
85
120
121
150
155
```

## Sample Output

```
Case 1: 40 80 120
Case 2: 40 80 120
```