1269 – Consecutive Sum

Little Jimmy is learning how to add integers. As in decimal the digits are 0 to 9, it makes a bit hard for him to understand the summation of all pair of digits. Since addition of numbers requires the knowledge of adding digits. So, his mother gave him a software that can convert a decimal integer to its binary and a binary to its corresponding decimal. So, Jimmy's idea is to convert the numbers into binaries, and then he adds them and turns the result back to decimal using the software. It's easy to add in binary, since you only need to know how to add (0, 0), (0, 1), (1, 0), (1, 1). Jimmy doesn't have the idea of carry operation, so he thinks that

1 + 1 = 0
1 + 0 = 1
0 + 1 = 1
0 + 0 = 0

Using these operations, he adds the numbers in binary. So, according to his calculations,

3 (011) + 7 (111) = 4 (100)

Now you are given an array of **n** integers, indexed from **0** to **n-1**, you have to find two indices **i j** in the array **$(0 \le i \le j < n)$**, such that the summation (according to Jimmy) of all integers between indices **i** and **j** in the array, is maximum. And you also have to find two indices, **p q** in the array **$(0 \le p \le q < n)$**, such that the summation (according to Jimmy) of all integers between indices **p** and **q** in the array, is minimum. You only have to report the maximum and minimum integers.

Input

Input starts with an integer **T (≤ 10)**, denoting the number of test cases.

Each case starts with a line containing an integer **n $(1 \le n \le 50000)$**. The next line contains **n** space separated non-negative integers, denoting the integers of the given array. Each integer fits into a 32 bit signed integer.

Output

For each case, print the case number, the maximum and minimum summation that can be made using Jimmy's addition.

Sample Input	Output for Sample Input

2
5
6 8 2 4 2
5
3 8 2 6 5
	Case 1: 14 2
	Case 2: 15 1

Note

Dataset is huge, use faster I/O methods.