

11717 Energy Saving Microcontroller

The rapid progress of the electronics era is mostly due to wide availability of microcontrollers. They are used in many devices, particularly in stand alone devices that need to do real time processing. Some common uses are in cell phones, hand held video games and pace makers. A lot of these stand alone devices run on internal battery power with no connection to external power source when in operation. Therefore it is critical that, the devices are able to minimize their power usage when idle. In this problem, you will be required to simulate the behavior of a simply designed microcontroller based device.

Input

The first line of input contains a positive integer T (≤ 10), where T denotes the total number of scenarios to simulate. Each scenario starts with three positive numbers n (≤ 1000), i and k . Both i and k will fit in **32 bit signed integer**. The next n lines contain a positive integer each. These numbers will be in increasing order and will fit in 32 bit signed integer. These numbers indicate the time in microseconds when an instruction is sent to the microcontroller for execution. If the microcontroller is in active state, it will execute the instruction instantly. In the event the microcontroller is inactive, it will take k microseconds to activate itself and then it will process the instruction. If any instructions are sent during the activation period, those will be ignored. The microcontroller is active at the 0-th microsecond, that is after it starts its operation. After that, if it doesn't execute any instruction for at least i microseconds, it goes into inactive state. Note that, the microcontroller may go into inactive state anytime it doesn't receive an instruction for at least i microseconds, after processing an instruction. If an instruction comes just when the device is about to go idle, the device will go idle and then the instruction will activate the device.

Output

For each input scenario, there will be one line of output. It will contain the case number followed by the number of times the microcontroller went into inactive state and the number of instructions that were ignored; the two numbers will be separated by a single space. Please look at sample output for exact formatting.

Sample Input

```
3
1 2 3
1
2 2 3
10
11
2 2 1
10
11
```

Sample Output

```
Case 1: 0 0
Case 2: 1 1
```

Case 3: 1 0