## Problem 5 – Change Even Bits

You are given a number **N** and you have to read from the console exactly **N + 1 integers** and additionally an integer **L** – the number to be processed. Your task is to count the bits in each of the **N** input integers (let this be the number **len**), then **change bit to "1" len even positions** (0, 2, 4, …) of the input number **L**. Print on the console the obtained number and the total number of bits that have **actually** been changed. To be counted as changed they should have been “0” at first and then changed to “1”. The comments in the example below will help you understand better what you should do. Note that "0" consists of 1 bit (**len**=1).

### Input

The input data should be read from the console.

* The first line holds an integer number **N** – the count of numbers to be read.
* The next **N** lines hold the **N input integers**.
* The last line holds an integer number **L** – the number to be processed.

The input data will always be valid and in the format described. There is no need to check it.

### Output

The output should be printed on the console. It should consist of **exactly 2** lines:

* On the **first** **line** print the **number L** after setting its **even** bits.
* On the **second** **line** print the number of bits that have **actually changed**.

### Constraints

* **N** will be an integer between 0 and 10.
* The **N** **input integers** in the input will be between 0 and 65535.
* **L** will be an integer number between 0 and 18 446 744 073 709 551 615.
* Allowed working time: 0.25 seconds. Allowed memory: 16 MB.

### Examples

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comments** |
| 2  22  15  240 | 501  3 | The first number **22** is **10110** in binary. It has **5** bits, so we have to set **5** even positions: **0, 2, 4, 6, 8**. The number **L** is **240** (**1111 0000**) before changing. After setting to "1" the positions 0, 2, 4, 6 and 8, we obtain **L** = **501** (**1** **1111 0101**). We have **actually** changed **3 bits**.  The second number **15** is **0000 1111** in binary. It has length **4** bits. The number **L** is now **501** (**1 1111 0101**) before changing. We have to set to "1" **4 even positions** (0, 2, 4, 6). After the changing, the number stays the same: **501** (**1** **1111 0101**), because all these positions already have value "1". We have **actually** changed **0 bits**.  The end result is **501**. The **total number of changed** bits is 3 + 0 = **3**. |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |
| 3  134  85  23  999 | 22519  4 |  | 2  155  155  309512 | 327005  6 |  | 1  0  0 | 1  1 | 4  65535  65535  65535  65535  0 | 1431655765  16 |