# Problem 5 – Wiggle Wiggle

Nakov likes Hip-hop music. He got inspired by the music clips he saw on YouTube and now he really wants to learn how to twerk. You got to help him to accomplish his goal. Since he is a lot into bits, you need to use bitwise operations.

You are given a **sequence of 64 bit integer** numbers on **one line,** separated by a space. You have to take each **consecutive pair** of numbers (first with second, third with fourth and so on). Then you have to pass through the bits from **right to left** and switch **every bit at even positions** from these two numbers. Assume the **position ‘0’ is even.**

For example, the first number pair is 36854775807 and 2285477580.

|  |  |
| --- | --- |
| 36854775807 | 000000000000000000000000000100010010100101101110011111111111111 |
| 2285477580 | 000000000000000000000000000000010001000001110011001111011001100 |

After switching, the numbers are:

|  |  |
| --- | --- |
| 36519493358 | 000000000000000000000000000100010000000101100110011111011101110 |
| 2554175453 | 000000000000000000000000000000010011100001111011001111111011101 |

In the end, you need to **reverse** the whole two numbers **bitwise**. The final result is:

|  |  |
| --- | --- |
| 9223372000335806737 | 111111111111111111111111111011101111111010011001100000100010001 |
| 9223372034233491490 | 111111111111111111111111111111101100011110000100110000000100010 |

### Input

The input data should be read from the console. It consists of a sequence of 64-bit integer numbers, separated by a space. The sequence **count** will always be an **even** number.

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

### Output

Print each of the obtained numbers on a single line in the following format:  
**{decimal} {binary}**

### Constraints

* **Every** **input number** will be a 64-bit integer in the range [0 … 9,223,372,036,854,775,807].
* The sequence **count** will always be an **even** number.
* Allowed working time for your program: 0.1 seconds.
* Allowed memory: 16 MB.
* Hint: You should print all **63** bits in the final result for each number (without the sign bit).

### Examples

|  |
| --- |
| **Input** |
| 36854775807 2285477580 |
| **Output** |
| 9223372000335806737 111111111111111111111111111011101111111010011001100000100010001  9223372034233491490 111111111111111111111111111111101100011110000100110000000100010 |

|  |
| --- |
| **Input** |
| 4558 488755 25 555 |
| **Output** |
| 9223372036854426212 111111111111111111111111111111111111111111110101010101001100100  9223372036854632089 111111111111111111111111111111111111111111111011100111010011001  9223372036854775798 111111111111111111111111111111111111111111111111111111111110110  9223372036854775236 111111111111111111111111111111111111111111111111111110111000100 |

|  |
| --- |
| **Input** |
| 5436 4266 64367 234256 423456 2 |
| **Output** |
| 9223372036854771671 111111111111111111111111111111111111111111111111110111111010111  9223372036854770241 111111111111111111111111111111111111111111111111110101001000001  9223372036854662341 111111111111111111111111111111111111111111111100100010011000101  9223372036854590650 111111111111111111111111111111111111111111111010010110010111010  9223372036854635999 111111111111111111111111111111111111111111111011101110111011111  9223372036854492157 111111111111111111111111111111111111111111110111010101111111101 |