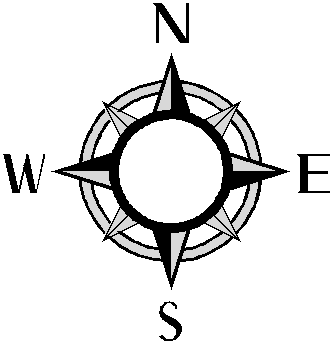
## Problem 5 – Formula Bit 1

The residents of Bitlandia are huge sports fans. The bits have played almost every single sport that they have learned from watching human TV i.e. EuroBitSport and BitTV. Today for the first time they watched Formula 1 and now they certainly want to build a local track and start practicing right away. Of course the bits don’t want to copy the humans. They want to be unique and therefore they’ve added some special rules:

1. The **track** must be built on a **grid** of **8x8 cells**, containing only zeros and ones.
2. The track itself must contain only **zeros**. The width of the track will be **one cell**.
3. The track must start from the **upper right corner** and end on the **lower left corner**.
4. The cars can move only in 3 directions – **South** (down), **West** (left) And **North** (up).
5. The first direction must always be **south**.
6. The cars must move in the current direction, while it is possible i.e. the cars can turn only when it reaches the **end** of the **grid** or a **cell**, containing a bit with a value of **one**.
7. The cars can switch between directions only in the following order:

South -> West -> North -> West (and then again South -> West and so on)

You will receive information about the grid as a list of **8** bytes (positive integers in the range [0…255]) n0, n1, …, n7. The grid itself is represented by the bits of those bytes.

Your task is to find whether a track can be built on the given grid. If the grid is appropriate, you should print the length of the track and the count of the turns in it (the switches between directions), otherwise you should print “No” and the length of the track until it was interrupted.

### Input

### The input data should be read from the console.

There will be exactly 8 lines each holding an integer number (n0, n1… n7).

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

### Output

The output data should be printed on the console.

On the only output row you should print two numbers in the following format “X Y”, where X is the length of the track and Y is the count of the turns. If a track cannot be built, you should print “No X”, where X is the length of the track, until it was interrupted.

### Constraints

* The numbers **n0**, **n1**, …, **n7** are positive integers in the range [0…255]
* Allowed work time for your program: 0.1 seconds.
* Allowed memory: 16 MB.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | *7* | *6* | *5* | *4* | *3* | *2* | *1* | *0* |  | | *0* | ***0*** | ***0*** | ***0*** | ***0*** | ***0*** | 0 | 1 | ***0*** | n0 = 2 | | *1* | ***0*** | 0 | 1 | 0 | ***0*** | 1 | 1 | ***0*** | n1 = 38 | | *2* | ***0*** | 0 | 0 | 1 | ***0*** | 1 | 0 | ***0*** | n2 = 20 | | *3* | ***0*** | 0 | 1 | 1 | ***0*** | ***0*** | ***0*** | ***0*** | n3 = 48 | | *4* | ***0*** | 1 | 1 | 0 | 1 | 1 | 1 | 1 | n4 = 111 | | *5* | ***0*** | 0 | 0 | 0 | 1 | 1 | 1 | 1 | n5 = 15 | | *6* | ***0*** | 0 | 0 | 0 | 0 | 0 | 1 | 1 | n6 = 3 | | *7* | ***0*** | 0 | 1 | 0 | 1 | 0 | 1 | 1 | n7 = 43 | | |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | *7* | *6* | *5* | *4* | *3* | *2* | *1* | *0* |  | | *0* | 0 | 0 | 0 | 0 | 0 | 0 | 1 | ***0*** | n0 = 2 | | *1* | 0 | 0 | 1 | 0 | 0 | 1 | 1 | ***0*** | n1 = 38 | | *2* | 0 | 0 | 0 | 1 | 1 | 1 | 0 | ***0*** | n2 = 28 | | *3* | 0 | 0 | 0 | 1 | ***0*** | ***0*** | ***0*** | ***0*** | n3 = 16 | | *4* | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | n4 = 71 | | *5* | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | n5 = 143 | | *6* | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | n6 = 3 | | *7* | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | n7 = 43 | |

### Examples

|  |  |
| --- | --- |
| **Example**  **input** | **Example**  **output** |
| 2  38  20  48  111  15  3  43 | 21 4 |

|  |  |
| --- | --- |
| **Example**  **input** | **Example**  **output** |
| 2  38  28  16  71  143  3  43 | No 7 |