## Problem 5 – Angry Bits

### The residents of Bitlandia love playing games. They try all sorts of games – outdoor games, team games, computer games, but their newest passion is the mobile gaming. All of the bits in Bitlandia compete in the game of Angry Bits. It’s a game played on a grid of 16x8 cells, containing zeros and ones. All of the ones on the left side (columns 8-15) denote a bird and all of the ones on the right side (columns 0-7) denote a pig. The object of the game is to destroy the pigs. To achieve this, the player throws the birds towards them. The player wins if he destroys all of the pigs. He also gets a score according to the destroyed pigs. There are some special rules:

1. There will be no more than one bird in each column on the left side. There could be zero birds in some columns though.
2. There could be any number of pigs in each column on the right side (maximum 8 per column).
3. The birds are thrown in a special order, from right to left i.e. first the bird from column 8, then the one from column 9 and so on (considering that there are birds in those columns).
4. Each bird flies diagonally upwards. When it reaches the top row, it starts flying diagonally downwards. Refer to the table. When the bird from cell [4, 8] is thrown it flies upwards to [0, 4], then starts flying downwards. (Note that if a bird is on the top row, it starts the flight diagonally downwards)
5. A hit encounters when a bird reaches a cell with a pig in it. When that happens, the pig is destroyed along with all other pigs from the neighbor cells (horizontally, vertically or diagonally). Following the previous example, when the bird hits the pig in [2, 2], the pig is eliminated along with all six pigs near it.
6. The player wins points after each hit. The score for the hit is calculated according to the following formula: Length of the flight \* Count of destroyed pigs. The length is measured in cells and does not include the starting cell. In our example the points for the hit are 42 (the flight is 6 cells long and the destroyed pigs are 7).
7. A bird ends its flight, when it hits a pig or when it reaches the right or bottom side of the field.
8. When a bird ends its flight, it is removed from the field along with all destroyed pigs i.e. their cells become zeros.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | *15* | *14* | *13* | *12* | *11* | *10* | *9* | *8* | *7* | *6* | *5* | *4* | *3* | *2* | *1* | *0* |  |
| *0* | 0 | 0 | **0** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **0** | 0 | 0 | 0 | 0 | n0 = 0 |
| *1* | 0 | **0** | 0 | **0** | 0 | 0 | 0 | 0 | 0 | 0 | **0** | 0 | **0** | 1 | 1 | 0 | n1 = 6 |
| *2* | **1** | 0 | 0 | 0 | **0** | 0 | 0 | 0 | 0 | **0** | 0 | 0 | 1 | **1** | 0 | 0 | n2 = 32780 |
| *3* | 0 | 0 | 0 | 0 | 0 | **0** | 0 | 0 | **0** | 0 | 0 | 0 | 1 | 1 | 1 | 0 | n3 = 14 |
| *4* | 0 | 0 | 0 | 0 | 0 | 0 | **0** | **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | n4 = 256 |
| *5* | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **0** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | n5 = 0 |
| *6* | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **0** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | n6 = 0 |
| *7* | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **1** | 0 | 0 | 0 | 0 | 0 | 0 | n7 = 64 |

### You will receive information about the game field as a list of 8 ushorts (positive integers in the range [0…65535]) n0, n1, …, n7. The grid itself is represented by the bits of those ushorts.

Your task is to find the total score of the player and whether he wins or not. The player wins if all pigs are destroyed.

### 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 the final score followed by the words **“Yes”** or **“No”** depending on whether the player wins or not. They should be separated by a single whitespace.

### Constraints

* The numbers **n0**, **n1**, …, **n7** are positive integers in the range [0…65535]
* Allowed work time for your program: 0.1 seconds.
* Allowed memory: 16 MB.

### Examples

|  |  |
| --- | --- |
| **Example**  **input** | **Example**  **output** |
| 0  6  32780  14  256  0  0  64 | 51 Yes |

|  |  |
| --- | --- |
| **Example**  **input** | **Example**  **output** |
| 0  1  0  0  0  0  0  265 | 8 No |