

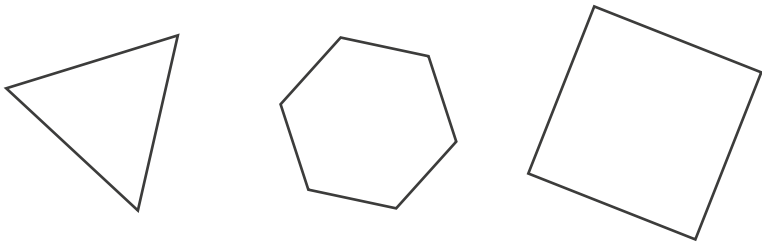
vi+aminlondon

Vitamin Puzzle  
Developer Test

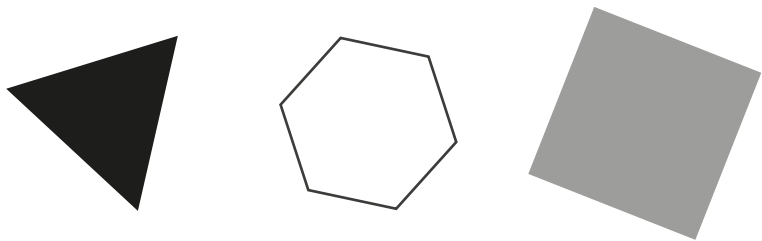


**Hello there, welcome to the Vitamin Puzzle**  
**Let's start with some basic information**

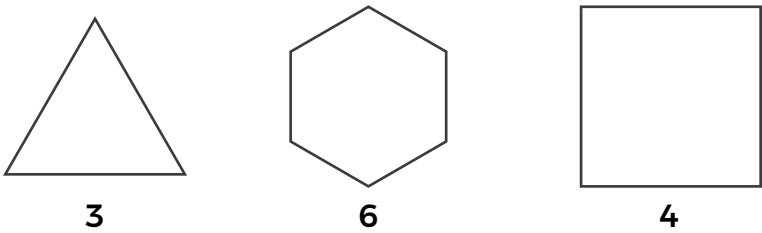
These shapes are called "**Vitamins**".



A Vitamin is therefore a convex regular polygon.  
These Vitamins have **colours**: white, grey or black.



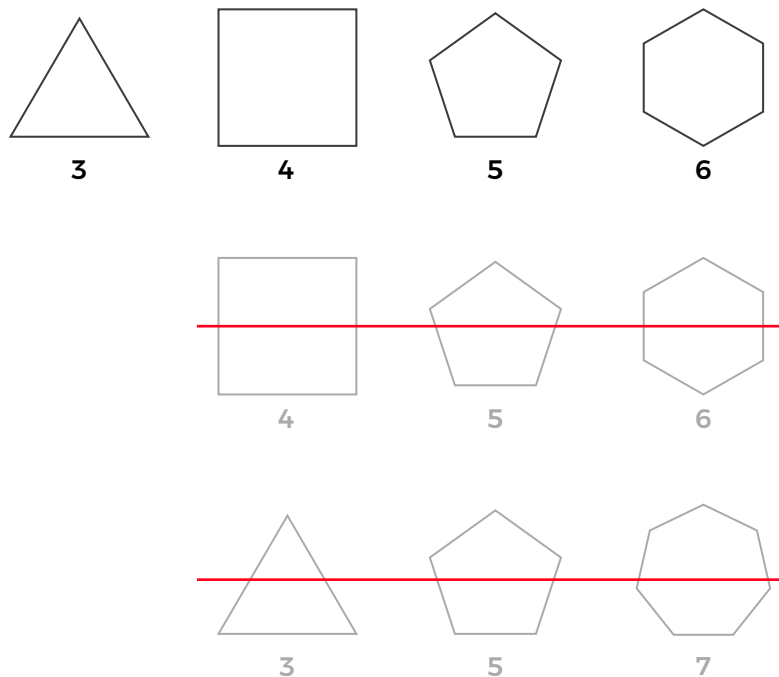
There exists a comparison operator that evaluates Vitamin value based on its number of edges/vertices.



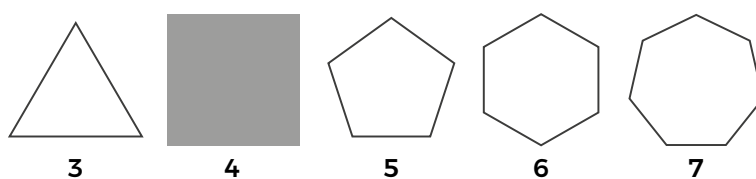
We can therefore say that:  
 $4 < 6$   
 $5 > 3$

## Now we've covered Vitamins, we can move on to Vitamin Lines.

A **Vitamin Line** is a finite length sequence of Vitamins arranged in a strictly ascending order, valued from 3 onwards with no gaps.



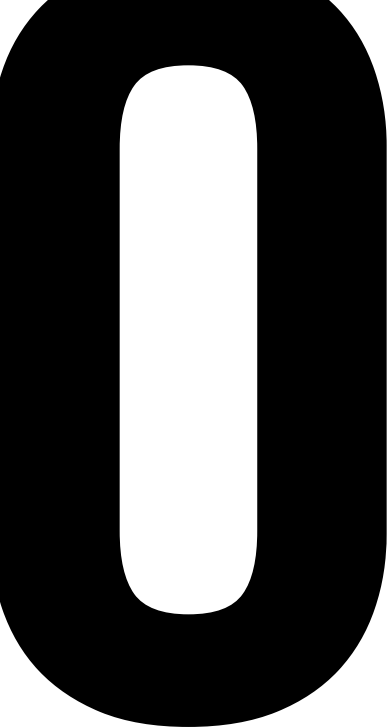
A Vitamin Line can be divided into three **colour groups**: white, grey and black. A colour group can be empty as well.



White group: 3, 5, 6, 7

Grey group: 4,

Black group: <empty set>



## To keep on learning, we have: Operations, Mini and Maxi Vitamins

In each non-empty colour group, there is always one **Mini-Vitamin** and one **Maxi-Vitamin**. A Mini-Vitamin is the smallest in its colour group, while a Maxi-Vitamin is the greatest in its colour group.

Based on the previous example, we have:

White Mini-Vitamin: 3

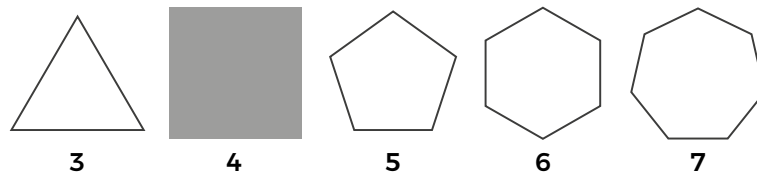
White Maxi-Vitamin: 7

Grey Mini-Vitamin: 4

Grey Maxi-Vitamin: 4

Black Mini-Vitamin: <none>

Black Maxi-Vitamin: <none>



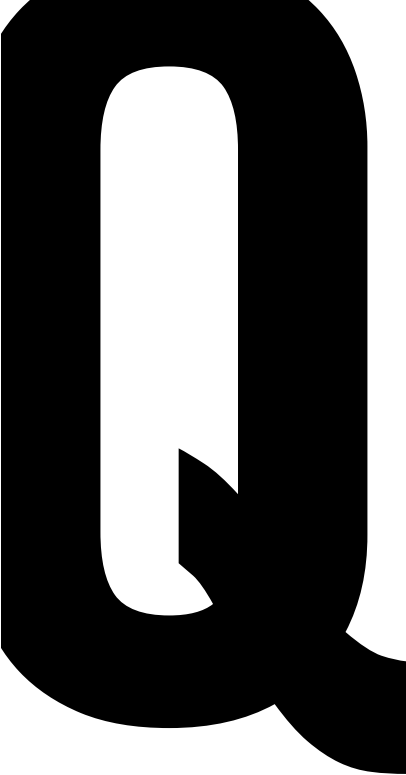
There is one atomic operation that can be performed in a Vitamin Line: **Colour Swap**.

The operation is performed on one Vitamin, and means it will change its colour (effectively going from one colour group to another).

The Colour Swap operation has two hard constraints:

- A. Only Maxi-Vitamins can perform the Colour Swap
- B. The changing Vitamin must remain a Maxi-Vitamin after the Colour Swap

They are known as the **Maxi-Maxi Principle**.



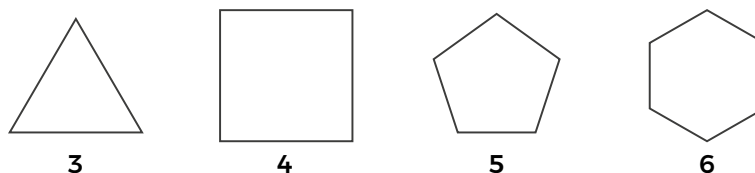
Got all that? Great.  
Time for some tasks.

### 1A

Given the starting situation of:



Please write a list of Colour Swaps to get to the goal situation:



### 1B

Write a function called `makeAllWhite` (in a programming language of your choice\*) that will take a string representing the initial situation as its argument and return the list of Colour Swaps (as a json string) that will allow for all Vitamins to become white. Input and output formats as below.

Input: `"3G 4G"` //means there are two grey Vitamins

Output:

```
[  
  [4, "G", "B"], //square Vitamin goes from grey to black  
  [3, "G", "W"], //triangular Vitamin goes from grey to white, etc.  
  [4, "B", "W"]  
]
```

Is your solution optimal?



**On to section two.  
Awesome stuff.**

### **2A**

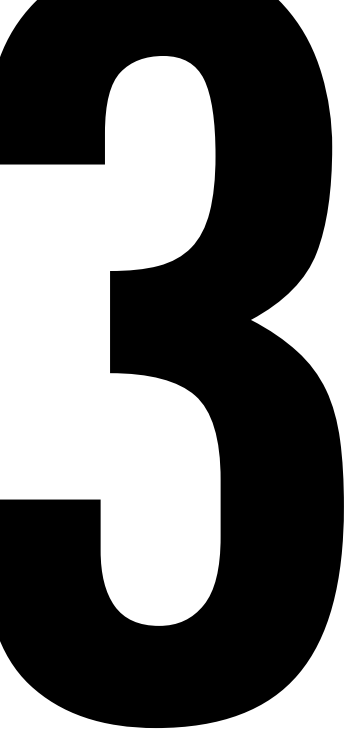
Write a piece of code that will prepare a visualisation of a Vitamin Line in a browser environment.

Input: "3B 4B 5G 6W" or similar.

### **2B**

Modify the code from TASK 2A so that if the input is provided again the visualisation will smoothly transit to the new state.

New input: "3G 4W 5G 6B" or similar.



**The finishing line is in sight.  
Still, more lines of code to write.**

### **3A**

Write a function that will take two arguments as input:

A. Initial state: "3G 4G"

B. List of operations:

```
[  
  [4, "G", "B"],  
  [3, "G", "W"],  
  [4, "B", "W"]  
]
```

And will produce a json-encoded array of consecutive states  
given the two:

(Output)

```
[  
  "3G 4G",  
  "3G 4B",  
  "3W 4B",  
  "3W 4W"  
]
```

### **3B**

Use the solutions for TASK 3A and TASK 2B to prepare  
animation consisting of multiple Colour Swap transitions.



**Finished, optimised and accurate?  
We're keen to see what you've created.**

When you have finished the tasks, please send the solutions to [piotr@vitaminlondon.com](mailto:piotr@vitaminlondon.com), cc'ing [jacob@vitaminlondon.com](mailto:jacob@vitaminlondon.com) and [richard@vitaminlondon.com](mailto:richard@vitaminlondon.com).

Please share your workings, do not worry if some of them did not yield (the correct) results. In case you got stuck half-way through, still include and send the notes across.

It all matters and we're keen to see!