# Find the Eggs



*After getting the colors and the eggs, it’s time for fun! Try and find the strongest eggs from the pile.*

Write a function called **find\_strongest\_eggs** that receives a **sequence** of eggs with their power and a **number** representing the number of **sub lists** that should be created from it.

**Each element** from the sequence should be placed **consecutively** in one of the sub lists. For example if you have a sequence **['a', 'b', 'c', 'd', 'e', 'f']** and you need to create 2 sub lists the result should be as follows: **[['a', 'c', 'e'],['b', 'd', 'f']]**.

Your job is to create the needed sub lists and follow the logic below to find whether the egg in the **middle of each sub list** is stronger than the rest. An egg is considered "stronger" if:

* Its value is **higher** than the values of the eggs to its **left and right**
* The value of the **right egg** (from the middle one) is **higher** than the value of the **left**. If we have more than one egg on the right/left side of the middle egg this rule should be followed for each **pair**.

***Note: Submit only the function in the judge system***

### Input

* There will be **no input**, just parameters passed to your function

### Output

* There is **no expected** output
* Thefunction should **return a list** with thestrongest egg **from each** sub list

### Constrains

* There won’t be any cases where the length of the sub lists is lower than 3 elements

### Examples

|  |  |  |
| --- | --- | --- |
| **Test Code** | **Output** | **Comment** |
| test = ([-1, 7, 3, 15, 2, 12], 2)  print(find\_strongest\_eggs(\*test)) | [3, 15] | We should create 2 sub lists:  **[-1, 3, 2]** => 3 is bigger than -1 **and** 2, and 2 is bigger than -1, so we return **3** as an egg which fullfills the above requirements  **[7, 15, 12]** => 15 is bigger than 7 **and** 12, and 12 is bigger than 7, so we return **15** |
| test = ([-1, 0, 2, 5, 2, 3], 2)  print(find\_strongest\_eggs(\*test)) | [5] | We should create 2 sublists ([-1, 2, 2], [0, 5, 3]) from the provided sequence and check whether each of them meets the given conditions. Since 2 is **not** bigger that 2 we don’t include it in the final result and continue with the second sub list. |
| test = ([51, 21, 83, 52, 55], 1) print(find\_strongest\_eggs(\*test)) | [83] | 83 is bigger than 52 **and** 21 so we continue to check the second condition.  52 is bigger than 21 and 55 is bigger than 51, so we consider 83 as stronger egg. |