# List Pureness



Write function called **best\_list\_pureness** which will receive a **list** **of numbers** and a number **K**. You have to **rotate** the list **K times** **(last becomes first)** to find the variation of the list with the **best pureness** (pureness is calculated by **summing** all the **elements** in the list **multiplied** by their **indices**). For example, in the list **[4, 3, 2, 6]** with the best pureness is **(3 \* 0) + (2 \* 1) + (6 \* 2) + (4 \* 3) = 26**. At the end the function should **return** a **string** containing the **highest pureness** and the **amount of rotations** that were made to find this pureness in the following format: **"Best pureness {pureness\_value} after {count\_rotations} rotations"**. If there is **more than one** highest pureness, take the **first one.**

### Input

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

### Output

* There is **no expected** output
* The function should **return a string** in the following format: **"Best pureness {pureness\_value} after {count\_rotations} rotations"**

### Examples

|  |  |  |
| --- | --- | --- |
| **Test Code** | **Output** | **Comment** |
| test = ([4, 3, 2, 6], 4)  result = best\_list\_pureness(\*test)  print(result) | Best pureness 26 after 3 rotations | Rotation 0 -> Pureness 25  Rotation 1 -> Pureness 16  Rotation 2 -> Pureness 23  Rotation 3 -> Pureness 26  Rotation 4 -> Pureness 25 |
| test = ([7, 9, 2, 5, 3, 4], 3)  result = best\_list\_pureness(\*test)  print(result) | Best pureness 78 after 2 rotations | Rotation 0 -> Pureness 60  Rotation 1 -> Pureness 66  Rotation 2 -> Pureness 78  Rotation 3 -> Pureness 78 |
| test = ([1, 2, 3, 4, 5], 10)  result = best\_list\_pureness(\*test)  print(result) | Best pureness 40 after 0 rotations |  |

*I love the way Earth rotates...  
It really makes my day.*