# 03. Naughty or Nice



*Santa Claus is always watching and seeing if children are good or bad. Only the nice children get Christmas presents, so Santa Claus is preparing his list this year to check which child has been good or bad.*

Write a function called **naughty\_or\_nice\_list** which will **receive**

* A **list** representing Santa Claus' "Naughty or Nice" list full of kids' names
* A **different number of arguments** (strings) **and/or keywords** representing commands

The function should **sort** the kids in the given Santa Claus list **into 3 lists**: **"Nice"**, **"Naughty"**, and **"Not found"**.

**The list holds a different number of kids - tuples** containing two elements: **a counting number** (integer) at the **first** position and **a name** (string) at the **second** position.

For example: **[(3, "Amy"), (1, "Tom"), (7, "George"), (3, "Katy")]**.

Next, the function could receive **arguments and/or keywords.**

Each **argument** is a **command**. The commands could be the following:

* **"{counting\_number}-Naughty"** - if there is **only one tuple** **in the given list** with the **same number**, **MOVE** the kid to a list with **NAUGHTY** kids and **remove** **it** from the Santa list. Otherwise, ignore the command.
* **"{counting\_number}-Nice"** - if there is **only one tuple** **in the given list** with the **same number**, **MOVE** the kid to a list with **NICE** kids and **remove** **it** from the Santa list. Otherwise, ignore the command.

Each **keyword** holds a **key** with a **name** (string),andeach **value** will be a string (**"Naughty"** or **"Nice"**):

* If there is **only one tuple** with the **same name**, **MOVE** the kid to a list with **NAUGHTY** or to the list with **NICE** kids depending on the **value in the keyword**. Then, **remove** **it** from the Santa list.
* Otherwise, ignore the command.

All **remaining tuples** in the given Santa's list are **not found kids**, and they should be **MOVED** to the **"Not found"** list.

In the end, **return the final lists, each on a new line as described below.**

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

### Input

* There will be **no input**. Just parameters passed to your function.

### Output

* The function should **return strings with the names on each list on separate lines**, **if there are any**, **otherwise skip the line:**
  + **"Nice: {name1}, {name2} … {nameN}"**
  + **"Naughty: {name1}, {name2} … {nameN}"**
  + **"Not found: {name1}, {name2} … {nameN}"**

### Examples

|  |  |
| --- | --- |
| **Test Code** | **Output** |
| print(naughty\_or\_nice\_list(  [  (3, "Amy"),  (1, "Tom"),  (7, "George"),  (3, "Katy"),  ],  "3-Nice",  "1-Naughty",  Amy="Nice",  Katy="Naughty",  )) | Nice: Amy  Naughty: Tom, Katy  Not found: George |
| print(naughty\_or\_nice\_list(  [  (7, "Peter"),  (1, "Lilly"),  (2, "Peter"),  (12, "Peter"),  (3, "Simon"),  ],  "3-Nice",  "5-Naughty",  "2-Nice",  "1-Nice",  )) | Nice: Simon, Peter, Lilly  Not found: Peter, Peter |
| print(naughty\_or\_nice\_list(  [  (6, "John"),  (4, "Karen"),  (2, "Tim"),  (1, "Merry"),  (6, "Frank"),  ],  "6-Nice",  "5-Naughty",  "4-Nice",  "3-Naughty",  "2-Nice",  "1-Naughty",  Frank="Nice",  Merry="Nice",  John="Naughty",  )) | Nice: Karen, Tim, Frank  Naughty: Merry, John |