## Photo Album

Create a class called **PhotoAlbum**. Upon initialization it should receive **pages** (**int**). It should also have **one more attribute**: **photos** (empty matrix) representing the album with its pages where you should put the photos. Each page can contain only **4 photos**. The class should also have **3 more methods**:

* **from\_photos\_count(photos\_count: int)** – creates a **new instance** of **PhotoAlbum**. Note: Each page can contain **4 photos**.
* **add\_photo(label:str)** – adds the photo in the **first possible page** and **slot** and return **"{label} photo added successfully on page {page\_number(starting from 1)} slot {slot\_number(starting from 1)}"**. If there are **no free slots** left, return **"No more free slots"**
* **display()** – returns a **string representation** of **each page** and the **photos** in it. Each photo is marked with **"[]"** and the **page separation** is made using **11 dashes (-)**. For example, if we have **1 page** and **2 photos**:

**-----------  
[] []  
-----------**  
and if we have **2 empty pages**:  
**-----------  
  
-----------  
  
-----------**

### Examples

|  |
| --- |
| **Test Code** |
| album = PhotoAlbum(2)  print(album.add\_photo("baby"))  print(album.add\_photo("first grade"))  print(album.add\_photo("eight grade"))  print(album.add\_photo("party with friends"))  print(album.photos)  print(album.add\_photo("prom"))  print(album.add\_photo("wedding"))  print(album.display()) |
| **Output** |
| baby photo added successfully on page 1 slot 1  first grade photo added successfully on page 1 slot 2  eight grade photo added successfully on page 1 slot 3  party with friends photo added successfully on page 1 slot 4  [['baby', 'first grade', 'eight grade', 'party with friends'], []]  prom photo added successfully on page 2 slot 1  wedding photo added successfully on page 2 slot 2  -----------  [] [] [] []  -----------  [] []  ----------- |