Problem 4. Write the following function. You may NOT hard code. (30 points)

Function Name: `pokeDex`

Inputs (2): (struct) 1xN structure array containing information about Pokémon
(cell) 1x(N+1) cell array containing information of a new field to be added

Outputs (1): (cell) 1xN updated structure array

Function Description:

Suppose you are given a 1xN structure array containing information about Pokémon. There are three fields in each structure: `Name`, `Stats`, and `Trainer`. The field `Name` represents the name of a Pokémon, and `Stats` contains a 1x1 structure with three fields: `HP`, `Attack`, and `Defense`. You are also given a cell array where the first element is the name of a new field to be added and the remaining elements correspond to each pokemon.

- First, use the cell array to add a new field to the structure array.
- Then, sort the structure array based on the attack level of each Pokemon (from smallest to largest). Use the `Attack` field from the nested structure in the `Stats` field.
- Next, delete any pokemon that is NOT a `'Dragon'` type.
- Finally, remove the field `Type` from the structure array.

`pokemon` →

`Name:`	**`'Dragonite'`**	`'Ekans'`	`'Ditto'`	**`'Kingdra'`**	**`'Flygon'`**
`Stats:`	**`1x1 struct`**	`1x1 struct`	`1x1 struct`	**`1x1 struct`**	**`1x1 struct`**
`Type:`	**`'Dragon'`**	`'Poison'`	`'Normal'`	**`'Dragon'`**	**`'Dragon'`**

Example of Dragonite's `Stats` structure →

`HP:`	`266`
`Attack:`	`275`
`Defense:`	`204`

Note:

Attack: Dragonite > Flygon > Kingdra

Example:

```
>> out = pokeDex(pokemon, {'Count', 1, 5, 1, 3, 8});
```

`Name:`	`'Kingdra'`	`'Flygon'`	`'Dragonite'`
`Stats:`	`1x1 struct`	`1x1 struct`	`1x1 struct`
`Count:`	`3`	`8`	`1`

Write your answer below:

SOLUTION:

```
function updated = pokeDex(info, cA)
% Add new field to structure array
% Extract out the attack level while I'm looping through the
% structure array
fNew = cA{1};
stats = [];
for i = 1:length(info)
    info(i).(fNew) = cA{i+1};
    stats = [stats info(i).Stats.Attack];
end

% Sort the structure array
[~, idx] = sort(stats);
info = info(idx);

% Delete non-dragon types
types = {info.Type};
mask = strcmp(types, 'Dragon');
info = info(mask);

% Remove the field Type from the structure array
updated = rmfield(info, 'Type');
end
```