**Assignment: Arrays**

Write a dynamic array class that has the following features:

1. Automatically allocate memory when items are added (in chunks).
2. Automatically deallocate memory when items are deleted (in chunks).
3. Track its own size using a class variable, independent of the actual memory allocated.
4. Initialize with 10 empty slots.
5. Increase or decrease the allocated memory slots by 5 at a time (i.e. chunk size is 5).

The API for your class is provided in **array\_api.py**. Please translate this API to your language, and maintain the public names and parameters whereever possible. To give you additional help, I've also left private methods in the API class; these can be deleted or modified as you wish.

**Getting Started**

I suggest that you create and test one method at a time as you work through add, insert, set, get, delete, and swap. Test your array class in a separate file named **main.py** (alter for your language).

For example, as your program the insert function, ensure it handles the following:

1. Inserting when there are empty slots in the internal array. Ensure all items above it are moved over one slot.
2. Inserting when there are no empty slots in the internal array. This results in a new internal array that has 5 additional slots (10 goes to 15, 15 goes to 20, etc.). After inserting the new value, the internal array should have four empty slots.
3. Inserting with an invalid index (< 0 or >= the current size).

Regarding this last point, your class should throw an exception when the index is out of bounds. This exception will be caught in your main program.

**Printing the Structure**

In order to grade your assignment, we need to be able to see the internal state of your array class. Add a debug\_print method to your class that prints the following line:

**11 of 15 >>> a, e, r, o, I, o, d, u, s, a, u, null, null, null, null**

In the line above:

* 11 is the current (logical) size of the array.
* 15 is the actual size of the allocated internal array. This number should always be in multiples of 5 (chunk size).
* a, e, r, o, and so on are the values in the array slots.
* null is an empty slot. This can be None, nil, null, or whatever word your language uses for null.

**The File of Instructions**

The assignment comes with a CSV file named **data.csv**. This file contains the instructions you should run on your Array class. It will assist in grading your work.

In your **main.py** file, read **data.csv** and interate through the commands. You can either use a CSV library in your language, or you can simply split each line by comma (there are no commas in the values). On each line, call the appropriate method in your Array class. The commands are as follows:

* CREATE creates an instance of your array class.
* DEBUG prints the debug line to the console.
* ADD adds a value to the end of the array.
* SET sets a value at a given index in the array.
* GET retrieves a value at a given index in the array. Your main program should print this value to the console.
* DELETE removes a value at the given index in the array. Be sure to shift all elements down to fill the empty slot.
* INSERT inserts a value at the given index in the array. Be sure to shift all elements up to make an empty slot.
* SWAP swaps two values at the given indices in the array.

Each time you run a command, print the zero-based file line number followed by the command, in this format:

LineNum:FileLine

Note that some commands in the file are meant to fail, such as trying to insert a value beyond the bounds of the array. You need to wrap each call in a try/catch exception block and print the following when this occurs:

Error: <msg from your class here>

See the example output below for a comprehensive list of what your output should look like.

**Example**

The **data\_example.csv** file contains an example set of instructions you can work with. I suggest you cut this file down to one or two instructions at a time as you develop rather than try to run the entire file at once.

I've included the output of my assignment in the **output\_example.txt** file so you can see exactly how your output should look. Note the following in the file:

* Line 1 shows an empty, initial array with 10 slots. The size of this array is 0, even though there are 10 slots allocated. The 0 value is kept in a class variable within your array class.
* Line 18 is a GET, so it has an 's' on the next line.
* Line 19 caused an error, so we see the exception text on the next line.

**Submitting the Assignment**

Zip the following files and submit on Learning Suite (again, using Python extensions here):

main.py, array.py, data.csv, output.txt

(add any additional files needed to run your program)