

0. Replace **999** with the last three digits of your student id and replace **nnn** with your full first name.
You must follow java naming convention, format your code properly, and use only meaningful comments (if any) to get full credits. Create a project named "int101final**999**".
1. Create a "Tool**999**" class in "util**nnn**" package as a utility class and implements the following public static methods:
 - 1.1. "isUsable**999**" method that receives an input argument of type String and returns a boolean.
It returns true if its input argument is not null and not a blank string.
 - 1.2. "isUsable**999**" method that receives an input argument of type Object[] and returns a boolean.
It returns true if its input argument is not null and is an array with length more than 0.
 - 1.3. "count**999**" method that receives an input argument of type Object[]
and returns an int, which is the number of objects in the array.
Note that if its argument is not usable
according to 1.2, return -1.
2. Create an "Item**999**" class in "valuablennn" package.
This class contains the following fields and methods:
 - 2.1. Two private fields: "name" (String, final) and "amount" (int).
 - 2.2. A private constructor that receives a name and an amount to set the values of the two fields respectively without checking the correctness of the arguments.
 - 2.3. A public static "create" method that receives two input arguments: "name" (String) and "amount" (int) respectively. This method returns a new Item**999** object with the given "name" and the given "amount" if the given "name" is usable according to 1.2 and the given "amount" is more than 0. Otherwise, this method returns null.
 - 2.4. An @Override public "toString" method that returns its content as a string in the following format: "(name,amount)".
 - 2.5. A public "add" method that receives an input argument: "item" (Item**999**).
Set the "amount" of this object to the summation of the two "amount" and returns the sum.
 - 2.6. A public "isMatched" method that receives an input argument: "item" of type Item**999**.
This method returns true if this object and the given "item" has the same "name".
Otherwise, this method return false.
3. Create an "Container**999**" class in "valuablennn" package.
This class contains the following fields and methods:
 - 3.1. A private final field: "items" which is an array of type "Item**999**" and is initialized to an array of size **999**.

- 3.2. An `@Override` public `"toString"` method that returns a string containing the number of `Item999` in the array (by using 1.3) followed by the string concatenation of `Item999`'s `toString()`.
- 3.3. A public `"add"` method that receives an input argument: `"item"` of type `"Item999"`. This method will put `"item"` in the array if `"item"` does not match any `"Item999"` in the array `"items"`. If `"item"` matches any `"Item999"` in the array, adjust that `"Item999"` in the array by `"item"`. If the `"item"` does not match any `"Item999"` in the array and there is no space left in the array to add `"item"`, this method returns `false`. Otherwise, it returns `true`.
- 3.4. A public `"remove"` method that receives an input argument: `"item"` of type `"Item999"`. This method will remove the `"Item999"` in the array (by replacing that slot with `null`) that matches `"item"` and returns `true`. Otherwise, it returns `false`.