Object Oriented Programming (OOP) Lab Task 03 E1

Create a class `**IntegerSet**` for which each object can hold integers in the range 0 through Size - 1. A set is represented internally as an `ArrayList` of ones and zeros.

- An `ArrayList` element `a[i]` is 1 if integer `i` is in the set.
- An `ArrayList` element `a[j]` is 0 if integer `j` is not in the set.

The class should have the following two private data members:

1. An `**ArrayList<Integer>**` to hold the set, created dynamically according to the specified size.
2. An integer to hold the maximum size of the set.

Provide the implementation of the following constructors and a destructor:

1. A **constructor** that accepts an integer representing the size of a set and initializes it to the "empty set," i.e., a set whose `ArrayList` representation contains all zeros.
2. A **copy constructor** to initialize a set object with an already existing object.

Provide the following member functions for the common set operations:

1. **`insertElement(int k)`:** Inserts a new integer `k` into the set by setting `a[k]` to 1.
2. **`deleteElement(int m)`:** Deletes an integer `m` by setting `a[m]` to 0.
3. **`printSet()`:** Prints a set as a list of numbers separated by spaces. Only prints elements that are present in the set (i.e., their position in the `ArrayList` has a value of 1). Prints "---" for an empty set.
4. **`unionOfSets(IntegerSet set1, IntegerSet set2)`:** Creates a third set that is the set-theoretic union of two existing sets. An element of the third set's `ArrayList` is set to 1 if that element is 1 in either or both of the existing sets, and an element of the third set's `ArrayList` is set to 0 if that element is 0 in each of the existing sets. The union is only possible if both sets have the same sizes.
5. **`intersectionOfSets(IntegerSet set1, IntegerSet set2)`:** Creates a third set which is the set-theoretic intersection of two existing sets. An element of the third set's `ArrayList` is set to 0 if that element is 0 in either or both of the existing sets, and an element of the third set's `ArrayList` is set to 1 if that element is 1 in each of the existing sets. The intersection is only possible if both sets have the same sizes.

6. **`findElement(int key)`:** Searches an integer `key` in a set and returns `true` if the key exists in the set, `false` otherwise.

Note: Please ensure that you handle the necessary validations and boundary checks in your implementation.