Lab 03 - Array Ordering Problems

Direction: Submit typed work in the Labs directory of your github repository, dropbox, or google classroom assignment. Each part should be a separate files. The files named should be `"lab3A.cpp"` and `"lab3B.h"` respectively.

Part A: In class

Your objective is to write a program that defines the following function

- ☐ Define an int function named `PlaceTarget()` that takes a generic array parameter, an int parameter, and a constant generic reference parameter respectively. Given that the int parameter represents the size of the array parameter and the elements of the array parameter are sorted in descending order, the function returns the index where the generic parameter would be if it were inserted in order.

Part B: Take home

Your objective is to define the generic class named **Bag** which represents a bag with a maximum capacity of 100 where a bag is a collection of object. For the class, you must include the following:

- ☐ A private generic array field of size 100 that represents the bag.
- ☐ A private int field that represents the current size of the bag.
- ☐ A public default constructor that assigns the default generic value to each element of the array field and assigns 0 to the int field.
- ☐ A public copy constructor.
- ☐ A public assignment operator.
- ☐ A public empty destructor.
- ☐ A public void method named `Add()` that takes a constant generic reference parameter. It adds the parameter to the bag.
- ☐ A public void method named `Remove()` that takes a constant generic reference parameter. It removes one of the element of the bag that equals to the parameter if the parameter is a member of the bag.
- ☐ A public bool constant method named `IsFull()` that takes no parameters. It returns true if the bag is full; otherwise, it returns false.
- ☐ A public bool constant method named `IsEmpty()` that takes no parameters. It returns true if the bag is empty; otherwise, it returns false.
- ☐ A public int constant method named `Count()` that takes no parameters. It returns the count of the bag.
- ☐ A public bool constant method named `Contains()` that takes a constant generic reference parameter. It returns true if the parameter is a member of the bag; otherwise, it returns false.
- ☐ A public constant method named `ToString()` that takes no parameters. It returns a string of elements of the bag separated by commas all enclosed in curly braces.
- ☐ A friend overloaded ostream operator. It displays the elements of the bag in the same format as `ToString()`.