Object Oriented Programming

Home Work 08 **Marks 10**

Instructions

Work on this home work individually. Absolutely NO collaboration is allowed. Any traces of plagiarism would result in a ZERO marks in this homework and possible disciplinary action. Tasks should be coded in **C++**.

Due Date

Paste the solution of the problem (source code .cpp file only) labeled with your complete roll number in SEM – HW 08 and SEA – HW 08 folders for SE Morning and SE Afternoon sections respectively on Tuesday, May 10, 2016 before 05:00 PM. These folders are available at \\printsrv\Teacher Data\Umair Babar\Students.

ADT: IntegerSet

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 array of **ones** and **zeros**.

- Array element **a[i]** is **1** if integer **i** is in the set.
- Array element **a[i]** is **0** if integer **i** is not in the set.

For Example; the following set contains values **0, 1, 3, 4, 7 and 9**.

0	1	2	3	4	5	6	7	8	9
1	1	0	1	1	0	0	1	0	1

The class should have following **two private data members**

1. An **integer pointer** that holds a reference of an array **created dynamically** according to the specified **size**.
2. A **constant integer** to hold the **maximum size** of the array.

Provide the implementation of following **constructors** and a **destructor**

1. A **constructor** which accepts an **integer** that represents the **size of a set** and initializes it to the so-called "empty set," i.e., a set whose array representation **contains all zeroes**.
2. A **copy constructor** to initialize a set object with already existing object.
3. A **destructor** to **free any memory resources** occupied by the **set** object.

Provide following member functions for the common set operations

1. **insertElement** that **inserts** a new integer **k** (passed as argument) into a **set** by setting **a[k]** to **1**.
2. **deleteElement** that **deletes** an integer **k** (passed as argument) by setting **a[k]** to **0**.
3. **unionOfSets** that **creates a third set** that is the set-theoretic **union of two existing sets** (i.e., an element of the third set's array 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 array is set to **0** if that element is **0** in each of the existing sets). *The union is only possible if both the sets have same sizes.*
4. **intersectionOfSets** which **creates a third set** which is the set-theoretic **intersection of two existing sets** (i.e., an element of the third set's array 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 array is set to 1 if that element is 1 in each of the existing sets). *The union is only possible if both the sets have same sizes.*
5. **findElement** that **searches** an integer **key** (passed as argument) in a **set** and return true if the **key** exists in the set, **false** otherwise.
6. **isNullSet** returns **true** if the set is an “empty-set” (a set whose array representation **contains all zeroes**), **false** otherwise.

Overload following operators

1. **Stream insertion (<<)** prints a **set** as a list of numbers separated by spaces. Print only those elements that are **present** in the set (i.e., their position in the array has a value of 1). Print - - - for an empty set.
2. **Assignment (=)** which copies the data of one object to another. The copy should only be done if both the objects have same sizes.
3. **Equal (==)** that determines whether **two sets are equal or not**. The operator should returns **true** if both the sets are equal, **false** otherwise.
4. **Logical NOT (!)** create and return a new **set** which contains the **reverse** of **left hand side object**, i.e. all the **1s** exist in the set converted to **0s** and vice versa.

Test the functionality of your created class by creating some of its objects. Also make calls to all the member functions of the class to test their functionality.

NOTE: - No submission will be accepted after the due date and time.

BEST OF LUCK