**CSC 331 Assignment 1**

In mathematics, **a set is a well-defined collection of distinct data. It is a special bag that does not allow repeated, or duplicate entries.** Specifications of the ADT Set is listed below.

|  |  |
| --- | --- |
| **ADT Set** | |
| **Data:**  A finite number of distinct objects with the same data type | |
| **Operations:** | |
| **Procedures** | **Description** |
| **getCurrentSize()** | Task: Gets the current number of entries in this set.  Input: None  Output: The integer number of entries currently in the set. |
| **isEmpty()** | Task: Sees whether this set is empty.  Input: None  Output: True if the set is empty, or false if not. |
| **add(newEntry)** | Task: Adds a new entry to this set, avoiding duplicates.  Input: newEntry, an object to be added as a new entry.  Output: True if the addition is successful, or false if the item already is in the set. |
| **remove(anEntry)** | Task: Removes one unspecified entry from this set, if possible.  Input: None.  Output: True if the removal was successful, or false if not. |
| **remove()** | Task: Sees whether this set is empty.  Input: None  Output: Either the removed entry, if the removal was successful, or null. |
| **clear()** | Task: Removes all entries from this set.  Input: None  Output: None |
| **contains(anEntry)** | Task: Tests whether this set contains a given entry.  Input: anEntry, the entry to locate.  Output: True if the set contains anEntry, or false if not. |
| **toArray()** | Task: Retrieves all entries that are in this set.  Input: None.  Output: A newly allocated array of all the entries in the set. |
| **union(setB)** | Task: perform set union operation () on this set(A) and a given set(B) .  Input: setB, a given set.  Output: The union of two sets ( no duplicates). |
| **intersection(setB)** | Task: perform set intersection operation() on this set(A) and a given set(B)  Input: setB, a given set.  Output: the difference of two sets ( no duplicates). |
| **difference(setB)** | Task: perform set difference operation(A - B) on this set(A) and a given set(B)  Input: setB, a given set.  Output: the difference of two sets ( no duplicates). |

1. Define a class **ArraySet** using an array that represents a set and implements the **ADT Set**. Make the **ArraySet** resizeable. Then write a C++ program that adequately demonstrates your implementation.
2. Define a class **LinkedSet** using a linked list that represents a set and implements the **ADT Set**. Then write a C++ program that adequately demonstrates your implementation.
3. Use the either **ArraySet** or **LinkedSet** to create a **spell checker**. The set serves as a dictionary and contains a collection of correctly spelled words, which can be read from an external file. To see whether a word is spelled correctly, you see whether it is contained in the dictionary. To simplify your task, restrict your dictionary to a manageable size.

Place the words whose spelling you want to check into a set, which can be read from another external file. The difference (set difference) between the dictionary (the set containing the correctly spelled words) and the set of words to be checked is a set of incorrectly spelled words.

Write a C++ program that adequately demonstrates your implementation.