HW4

W3101: Programming Languages, C++: Homework 4

- **Due Oct 17.**
- Please submit your solutions in Courseworks, do not mail them to me.
- Please submit ONLY your source (.cpp) files, not the entire project.
- Please call your files < YourName >_HW_4_1.cpp, and < YourName >_HW_4_2.cpp to help TA identify your files quickly.

1. Set is a very useful data structure in computer science. It is like an array, except that there is no ordering of the elements. That is, it stores multiple elements of the same type, but they need be stored (or retrieved) in an ordered fashion like an array. In this homework, implement a template class called "MySet". It should take a generic type "T" and should have the following:
   I. A private variable called "values", which is an array of type "T". (You can assume there will not be more than 100 elements (to allocate array size). That is, you can preallocate a size of 100 elements, if you need to.)
   II. A private variable called "count" of type int that keeps track of the number of elements in values.
   III. A public constructor that sets count to 0.
   IV. A public function called "void insert (T val)" that inserts the element "val" into the array values and increments "count".
   V. A public function called "bool findValue (T val)" that returns true if "val" is present in the array "values", and false if it is not present.
   VI. Do the following in the main function:
      i. Create an object of "MySet" with type "string".
      ii. Ask the user to enter a string. Insert it into the set using the "insert(..)" function. Repeat the process (of prompting the user to enter a string and storing it into the set) until the user enters "DONE".
      iii. Ask the user to enter another string. Find out if the string is present in the set using the "findValue(..)" function. If it is present, print that the string is preset in the set, and not present if it is not.

   Note: You are implementing "MySet" with an array. The actual Set data structure uses some other implementation (not an array).

2. Map is a very useful data structure in computer science. As the name implies, it maps "keys" to "values". One example is "name" to "phone number". That is, given a name (of type string), you can map it to a phone number of type int (or long). In this homework, implement a template class called "MyMap". It should take two generic types "K" for keys and "V" for values. It should have the following:
   I. A private variable called "keys", which is an array of type "K". (You can assume there will not be more than 100 elements. That is, you can preallocate a size of 100 elements, if you need to.)
   II. Another private variable called "values", which is an array of type "V". You can assume there are no more than 100 elements (to define array size).
   III. A private variable called "count" of type int that keeps track of the number of elements in keys and values.
   IV. A public constructor that sets count to 0.
   V. A public function called "void insert (K key, V val)" that inserts the element "key" to keys, "val" into the array values, and increments "count".
   VI. A public function called "bool findValue (K key, V& val)". This function should set "val" (passed as argument) to the value corresponding to "key" if "key" is present in the array "keys".

VII. A public function called "bool findKey (K& key, V val)". This function should set the variable key (passed as argument) to the key corresponding to "value" if "val" is present in the array "values".

VIII. Do the following in the main function:

   i. Create an object of "MyMap" with type "string" and "int" representing the name and phone number of a user.
   ii. Ask the user to enter a string and then an integer. Insert them into the map using the "insert(..)" function. Repeat the process (of prompting the user to enter string and int and storing them into the map) until the user enters "DONE".
   iii. Ask the user to enter another string. Find out if the string (key) has a corresponding value in the map using the "findValue(..)" function. If it is present, print that the corresponding integer for the string. If the string is not present, print "not present".
   iv. Next, ask the user to enter an integer. Find out if the integer (value) has a corresponding key in the map using the "findKey(..)" function. If it is present, print that the associated string (key) for the integer. If the value is not present, print "not present".

Note: You are implementing "MyMap" with two arrays. The actual Map data structure uses some other implementation (not arrays).

Ramana Isukapalli