1- Declare and implement a class ***fraction*** that has two data members: ***num*** and ***denom***, (both of type **int**), which represent the fraction's numerator and denominator, respectively. It has two member functions: ***getFract()*** that reads the ***num*** and ***denom*** of a ***fraction*** object from the keyboard in fractional form (i.e. num/denom), and ***showFract()*** that displays a ***fraction*** object in fractional form. It also has two overloaded operators + and *, for performing the addition and multiplication operations on any two objects of class ***fraction***, respectively. Note that, the formulas for adding and multiplying two fractions a/b and c/d are as follows:

$$\frac{a}{b}+\frac{c}{d}=\frac{a*d+b*c}{b*d}, \qquad \frac{a}{b}\times\frac{c}{d}=\frac{a*c}{b*d}$$

Write a program that declares two objects of class ***fraction***, and repeatedly reads data for these two objects from the keyboard, then perform the two operations on them, displays the results, and asks the user whether he/she wants to continue or not.

2- Declare and implement a class ***Date*** that has 3 **int** data members: day, month and year, and has a no-argument **constructor** that creates a ***Date*** object and initializes its data members to zeros, a 3-argument **constructor** that creates a ***Date*** object and initializes its data members to given values, and two member functions: ***get_date()*** that reads from the keyboard the data of a ***date*** object in the form dd/mm/yy, and ***show_date()*** that displays a ***date*** object in the form dd/mm/yy, and a **operator** < that compares two date objects and returns true if the first date is less than the second one, and false otherwise.

Finally, write a main program that creates a ***date*** object and initializes it with today's date, and creates another ***date*** object and reads its data from the keyboard, then compares these two dates and displays them with a message indicating the result of the comparison.

3- **Part A:**

A set is a collection of values; the order of the values in the set is irrelevant, and duplicate values are not allowed. Define a class **set**, whose objects are sets of integers. Class **set** has two data members: ***elem*** (array of integers) that holds the elements of the set, and ***size*** (of type int) that holds the number of elements in the set. Class **set** has a constructor ***set()***, which creates an empty set, a member function ***is_element()***, which returns true if a given value belongs to a set object, a member function ***empty()***, which returns true if a set object is empty, a member function ***show_set()***, which displays the elements of a set object. Also, class **set** has the following operators:

- **Operator +** yields the *union* of two sets, which contains those elements that belong to either or both sets.
- **Operator –** yields the *difference* of two sets, which contains those elements that belong to the first set but not to the second.
- **Operator *** yields the *intersection* of two sets, which contains those elements that belong to both sets.
- **Operator =** copies a given set into another one.
- **Operator +=** adds a given value to the set if it does not already belong to it.
- **Operator –=** removes a given value if it belongs to the set.

- **Operator** < compares two sets and returns true if the first set is a subset of the second, i.e. if every element of the first set also belongs to the second.
- **Operator** > compares two sets and returns true if the first set includes the second, i.e. if every element of the second set also belongs to the first.
- **Operator** == compares two sets and returns true if they have the same elements.

Part B:

Using the class *set*, write a main program that performs the following tasks:

1. Create two objects, set1 and set2, of the class set.
2. Read the elements of the two sets: set1 and set2.
3. Find the union of the two sets, store it in another set object and display it.
4. Find the differences (set1 – set2) and (set2 – set1), store them in two set objects and display them.
5. Compare the two sets and display one the following messages accordingly:
   - The two sets are identical.
   - The first set is a subset of the second set.
   - The first set includes the second set.
   - The two sets are not identical.
6. Find the intersection of the two sets, and store it in another set object. If the intersection is not empty, display it, and remove the common elements from both sets, then display the resultant two sets.

Run your program with different data to cover all the possibilities to make sure that all the set functions and operators work properly.

30 - 10 - 2011

1 - 9 - 2020

[illegible] - 8 - [illegible]

10 - 8 - 2020