Computer Programming 2 - Assignment 3

1) Write a template for a function called total. The function should calculate the sum of all members of an array and return the total. The arguments sent into the function are the array and the number of elements. Test the template in a simple main function that sends arrays of integers and doubles as arguments and displays the results.
Save the program as Q1.cpp

2) Design a class called Date. The class should store a date in three integers: month, day, and year. There should be accessor and mutator functions and a default constructor that sets the date to January 1, 2000. Also, write one input member function, and three output member functions (named output1, output2, and output3) to print the date in the following formats, respectively:

12/25/2014
December 25, 2014
25 December 2014

Input Validation: In your input function ask for month first, do not accept values for the month greater than 12 or less than 1. Do not accept values for the day less than 1 or greater than the number of days in the month.

Define a member function named Increment to display the next day.
Define a member function named Decrement to display the previous day

The class should detect the following conditions and handle them accordingly:
• When a date is set to the last day of the month and incremented, it should become the first day of the following month.
• When a date is set to December 31 and incremented, it should become January 1 of the following year.
• When a day is set to the first day of the month and decremented, it should become the last day of the previous month.
• When a date is set to January 1 and decremented, it should become December 31 of the previous year.
Demonstrate the class's capabilities in a simple program.

Number of days in January, March, May, July, August, October, and December: 31
Number of days in April, June, September, October: 30
Number of days in February: 28 (no need to consider leap years in your program)
Save the program as Q2.cpp

3) Suppose an airline company has hired you to write a program for choosing the passengers who should leave an overbooked flight. That means the airline has booked too many passengers hoping some of them would not show up on time, but passengers are on board and

the aircraft doesn't have enough seats available for all these passengers. Your program chooses the passenger who must leave the flight.

Passengers will be ranked based on the fare they have paid for booking (a double number called fare), and their number of loyalty points (an integer number called points). Fare and points are equally important, that means, in order to decide the position of each passenger in the list, fare and points each have 50% importance. A Person with minimum priority (combination of fare and points) will leave the flight.

You need to define a class called Passenger with these member variables:

- passenger's name
- fare
- points.

Passenger class must have 3 accessors, 3 mutators, a constructor with three parameters, an input function to read the passenger's info, and an output function to display the passenger's info.

Also, you need a class called Flight, in which you store a string for the flight number and the information of all passengers on any flight (a dynamic array of Passenger objects). In the Flight class, you need a function called insert (for adding a passenger), display (to display the name, fare, and points of all passengers in ascending order), and leave (for choosing the passenger who leaves the aircraft and removing his/her info from the passenger list). **It is important that the display function displays passengers in ascending order.** So, the person on the top of the list is the one who will leave first, not the passenger who booked the flight first and was added to the list of passengers before others.

Finally, write a main function that tests your classes and displays the following output. you need to add 5 passengers (not necessarily in the order shown below), display the list of passengers (you see the list is in ascending order), remove the first passenger, display the list again, remove another passenger, and display the list again.

This type of list of passengers is called a Priority Queue. This is not the same as Queue you learned in this course. Regular Queues are first in first out (FIFO). But this list of passengers is "paid less first out". Passengers who paid less fare (purchased their ticket on sale) or passengers who fly less frequently are considered as passengers with lower priority and must leave before others.

Save the program as Q3.cpp

```
Queue is :
 Fare         Points       Name
--------------------------------------
490.28         220         Terry
480.12         245         John
698.95         100         Alex
577.21         311         Susan
590.08         414         Brian

Customer who leaves is: Terry
Queue is :
 Fare         Points       Name
--------------------------------------
480.12         245         John
698.95         100         Alex
577.21         311         Susan
590.08         414         Brian

Customer who leaves is: John
Queue is :
 Fare         Points       Name
--------------------------------------
698.95         100         Alex
577.21         311         Susan
590.08         414         Brian
```