Assignment 8 Design

class Sort
Public: data[] : int temp: int length: int

class BubbleSort :class Sort
Public: Bubble() :void

class QuickSort : class Sort
Public: Qsort(int a[],int low,int high) :void

class RecursiveBinarySearch
Public: Search(int b[],int left,int right,int key) : int

Design:

1. Set sort class which include an int array data[], int temp which is the template number during swapping, int length is the length of the array.
2. Create a BubbleSort class as a child class of Sort.
3. BubbleSort(), I use 2 loop to build bubble sort function ,the 1st loop form 1 to the last element , the second loop is inside the 1st loop and from the i (which is the loop number of 1st loop)to the last of the element. Then use if loop and temp element to swap the 2 numbers. If a is smaller than b, do swapping or a is greater than b, continue to next loop or return.
4. Create a QuickSort class as a child class of Sort.
5. Qsort(int a[],int low,int high), in this function ,I used recursion.1st we need to set the condition of return, I chose when low is greater than high, because in the process of recursion, low will be +1 every time

and high will -1 every time. Then build the while loop to sort the number. In the end, use 2 Qsort() function to sort all the element in the array.

6. Create a RecursiveBinarySearch class.
7. Search(int b[],int left,int right,int key), the principle are the same with Qsort because the recursion. We choose the middle one number every time, if middle number is 0 then return true, if it is not 0 go in to the recursive function do it again, and use recursion to return the function. If we cannot find 0 in this array, in the end of this loop, return false.
8. Main(),the main aim is to find the integer array which fill in numbers. So 1st I use getline and istringstream o transfer each element to a string type array , 2nd I use atoi function to change the type from string to int, in the end use Qsort Search to get the result.

Test:

Input	Expect
1 2 3 4 6 5	false 1 2 3 4 5 6
-8 -9 -3 -5 -2	false -9 -8 -5 -3 -2
2 5 6 0 3 7	true 0 2 3 5 6 7
852 -1 0 2 3 963	true -1 0 2 3 852 963
1004 2004 -638 2	false -638 2 1004 2004