

## Bubble Sort Assignment

Sorting is the process of taking items of the same type and putting them in some defined order.

The **bubble sort** is an algorithm that compares adjacent items and swaps those that are not in order. The process is repeated until the list is in order.

**The number of passes is one less than the number of items.** In the example below, 5 names are being sorted therefore 4 passes will be required to complete the sort.

The **bubble sort** compares the 1<sup>st</sup> and 2<sup>nd</sup> values and swaps them if not in order, it then compares the 2<sup>nd</sup> and 3<sup>rd</sup> values.....continuing the process until the next-to-last and last values are compared.

Pass #1				
<b>Mary</b>	Fred	Fred	Fred	Fred
<b>Fred</b>	<b>Mary</b>	Mary	Mary	Mary
Tom	<b>Tom</b>	<b>Tom</b>	Ben	Ben
Ben	Ben	<b>Ben</b>	<b>Tom</b>	Sam
Sam	Sam	Sam	<b>Sam</b>	Tom

The final column becomes the first column for the next pass.

Pass #2				
<b>Fred</b>	Fred	Fred	Fred	Fred
<b>Mary</b>	<b>Mary</b>	Ben	Ben	Ben
Ben	<b>Ben</b>	<b>Mary</b>	Mary	Mary
Sam	Sam	<b>Sam</b>	<b>Sam</b>	Sam
Tom	Tom	Tom	Tom	Tom

Pass #3				
<b>Fred</b>	Ben	Ben	Ben	Ben
<b>Ben</b>	<b>Fred</b>	Fred	Fred	Fred
Mary	<b>Mary</b>	<b>Mary</b>	Mary	Mary
<b>Sam</b>	<b>Sam</b>	<b>Sam</b>	<b>Sam</b>	Sam
Tom	Tom	Tom	<b>Tom</b>	Tom

Pass #4				
<b>Ben</b>	Ben	Ben	Ben	Ben
<b>Fred</b>	<b>Fred</b>	Fred	Fred	Fred
<b>Mary</b>	<b>Mary</b>	<b>Mary</b>	Mary	Mary
<b>Sam</b>	Sam	<b>Sam</b>	<b>Sam</b>	Sam
Tom	Tom	Tom	<b>Tom</b>	Tom

After one pass the bottom value is in its correct position and will not have to be swapped. The next pass will not have to check the bottom value – each successive pass will have one less comparison to perform. A more efficient bubble sort will be explored later.

## Assignment

1. Sort the following set of values using a bubble sort. Numbers should be placed in numerical order (smallest to largest). Show all steps.

Pass #1				
6				
3				
5				
9				
2				

Pass #2				

Pass #3				

Pass #4				

2. A bubble sort requires the interchange (the swapping) of values stored in an array. Write the code needed to perform one of these swaps between values stored in an array at the following locations  $name(n)$  and  $name(n+1)$ . The swap will only occur if the value stored in  $name(n)$  is greater than  $name(n+1)$ .

3. Write the code to initialize the following list of names in an array called *name*.

“Joe”, “Mary”, “Sam”, “Tim”, “Ben”, “Harry”, “Michael”, “Anne”, “Kim”, “Lebron”, “Duane”, “Jeff”, “Julie”, “Karen”, “Kristin”

4. Write a program to alphabetically sort names in the name array initialized in question 3. The unsorted list of names should be displayed in a listbox, followed by the sorted list. A For..Next loop should be used for this assignment.