**Bubble Sort**  
Bubble Sort is probably one of the oldest, most easiest, straight-forward, inefficient sorting algorithms. It is the algorithm introduced as a sorting routine in most introductory courses on Algorithms. Bubble Sort works by comparing each element of the list with the element next to it and swapping them if required. With each pass, the largest of the list is "bubbled" to the end of the list whereas the smaller values sink to the bottom. It is similar to selection sort although not as straight forward. Instead of "selecting" maximum values, they are bubbled to a part of the list. An implementation in C.

void BubbleSort(int a[], int array\_size)  
{  
 int i, j, temp;  
 for (i = 0; i < (array\_size - 1); ++i)  
 {  
 for (j = 0; j < array\_size - 1 - i; ++j )  
 {  
 if (a[j] > a[j+1])  
 {  
 temp = a[j+1];  
 a[j+1] = a[j];  
 a[j] = temp;  
 }  
 }  
 }  
}

**Selection Sort**  
The idea of Selection Sort is rather simple. It basically determines the minimum (or maximum) of the list and swaps it with the element at the index where its supposed to be. The process is repeated such that the nth minimum (or maximum) element is swapped with the element at the n-1th index of the list. The below is an implementation of the algorithm in C.

void SelectionSort(int a[], int array\_size)  
{  
 int i;  
 for (i = 0; i < array\_size - 1; ++i)  
 {  
 int j, min, temp;  
 min = i;  
 for (j = i+1; j < array\_size; ++j)  
 {  
 if (a[j] < a[min])  
 min = j;  
 }  
  
 temp = a[i];  
 a[i] = a[min];  
 a[min] = temp;  
 }  
}

**Insertion Sort**  
The Insertion Sort algorithm is a commonly used algorithm. Even if you haven't been a programmer or a student of computer science, you may have used this algorithm. Try recalling how you sort a deck of cards. You start from the begining, traverse through the cards and as you find cards misplaced by precedence you remove them and insert them back into the right position. Eventually what you have is a sorted deck of cards. The same idea is applied in the Insertion Sort algorithm. The following is an implementation in C.

void insertionSort(int a[], int array\_size)  
{  
 int i, j, index;  
 for (i = 1; i < array\_size; ++i)  
 {  
 index = a[i];  
 for (j = i; j > 0 && a[j-1] > index; j--)  
 a[j] = a[j-1];  
  
 a[j] = index;  
 }  
}