Search and Sort in One dimensional array

- **Search** : linear search vs. binary search (requires the array elements to be sorted)
    - return the subscript of the array element that match the value that is being searched for
    - return -1 if the value is not there

- **Linear search**

```
int LinearSearch (const int a[], int aSize, int toFind)
{
    // Look through all items, starting at the front.
    for (int i = 0; i < aSize; i++)
        if (a[i] == toFind)
            return i;

    // You've gone through the whole list without success.
    return -1;
}
```

- **Binary search**

```
 int BinarySearch(int a[], int aSize, int toFind)
{
   int start = 0;                          //the search starts with index 0
   int last = aSize -1;                    //last is the last array index

   while (start <= last)                   //while there is still a place to look.
   {
      int middle = (start + last) / 2;     //Look here first
      if (toFind == a[middle])             //Found item. Quit.
           return middle;
      if (toFind > a[middle])              //Look in the last half
           start = middle + 1;
      else                                 //OR look in the first half
           last =  middle - 1;
   }

  //the element wasn't found
   return -1;
}
```

- **Sorting**
  (http://www.algolist.net/Algorithms/Sorting/Selection_sort)

```
void SelectionSort(int arr[], int n) {

    int i, j, minIndex, tmp;

    // repeat pair-wise comparison across the elements n-1 times
    for (i = 0; i < n - 1; i++) {

         // find the index of the element with the smallest value in the remaining elements
        minIndex = i;
        for (j = i + 1; j < n; j++)  {
            if (arr[j] < arr[minIndex])
                minIndex = j;
         }

        if (minIndex != i) {

           // swap arr[i] and arr[minIndex]
            tmp = arr[i];
            arr[i] = arr[minIndex];
            arr[minIndex] = tmp;
        }
    }
}
```

You may also write a separate function to swap the values of arr[i] and arr[minIndex]

```
void Swap (int  &value1,  int  &value2)
{
        int    tmp;

        tmp = value1;
        value1 = value2;
        value2 = tmp;

        return;
}
```