// Quicksort

```
#define IN
#define OUT
...

void Quicksort( IN OUT int v[], IN int n ) {
  int pivotIndex;

  if ( n > 0 ) {
    Partition( v, n, pivotIndex );
    Quicksort( v, pivotIndex );
    Quicksort( v+pivotIndex+1, n - pivotIndex-1 );
  }
}

void Partition( IN OUT int v[], IN int n, OUT int& pivotIndex ) {
  int pivot = v[0];
  int left = 1;         // 1st index after pivot
  int right = n-1;

  while ( left <= right ) {

    // move the left index towards the right
    while( left <= n && v[left] <= pivot )
      left++;

    // v[0],...v[left-1] <= pivot
    // v[left] > pivot

    // move the right index towards the left
    while ( v[right] > pivot )
      right--;

    // v[right] <= pivot
    // v[right+1],...,v[n-1] > pivot

    if ( left < right ) {

      // swap v[left] with v[right]
      int swap = v[left];
      v[left] = v[right];
      v[right] = swap;
    }
  }

  // swap v[0] with v[right]
  v[0] = v[right];
  v[right] = pivot;
  pivotIndex = right;
}
```