Translate the following quick sort algorithm to MIPS assembly language. (YOU ARE TO USE THE MIPS CALLING CONVENTIONS WHEN IMPLEMENTING THE QUICK SORT, PARTITION, AND SWAP SUBPROGRAMS!!!)

```assembly
main( )
begin
    quickSort(array, 0, n-1)
end main

quickSort( start address of array with n integers, integer low, high)
begin
    if (low < high) then
        pivotPosition = partition(array, low, high)
        quickSort(array, low, pivotPosition-1)
        quickSort(array, pivotPosition+1, high)
    end if
end quickSort

function partition( start address of array with n integers, integer low, high ) returns an integer
begin
    local variables
        integer left, right, pivot
    begin
        left = low
        right = high
        pivot = array[low]
        while left < right do
            while array[right] > pivot do
                right = right - 1
            end while
            while (left < right) and (array[left] <= pivot) do
                left = left + 1
            end while
            if left < right then
                swap(array[left], array[right])  //pass addresses
            end if
        end while
        array[low] = array[right]
        array[right] = pivot
        return right
    end partition

swap( address of integer operand1, address of integer operand2)
begin
    local variable
        integer temp
    begin
        temp = operand1
        operand1 = operand2
        operand2 = temp
    end swap

```

Turn in the following:
1) The MIPS assembly language program -- include comments describing which registers are being used for parameters and local variables
2) The "output" of the MIPS program using the following 11 integers for your initial data:

```
8   2   7   10   5   6    3    4    9     3    1
```

(I want a screen shot of the MIPS simulator including the sorted array in memory after the program has run).