

## (CSCI 421 VA) Project Two Implement Quick Sort Algorithm

### **Problem Description**( Refer to Exercise 6 on Page 116)

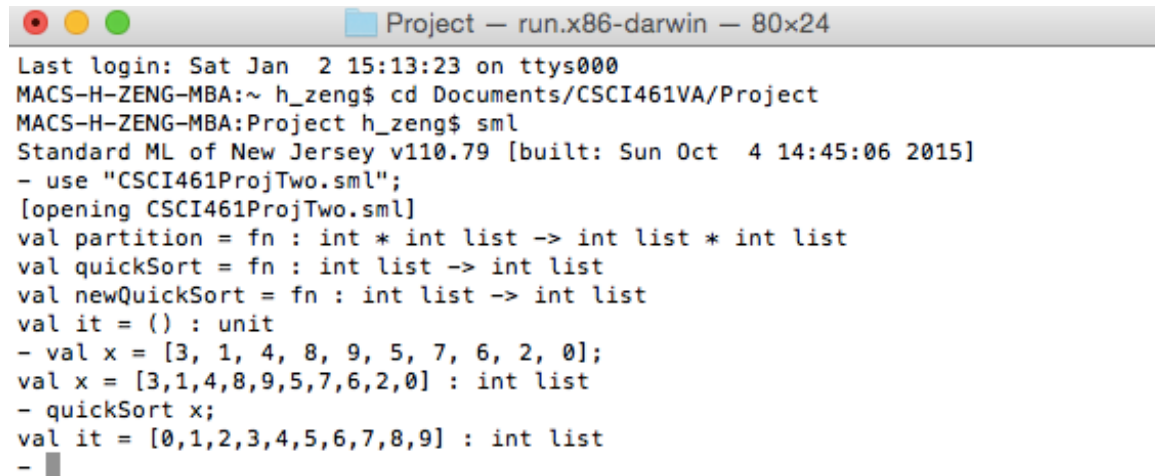
Write a quick sort function of type `int list -> int list`. Here is a review for quick sort algorithm. First pick an element and call it pivot. (The head of the list is an easy choice for the pivot.) Partition the rest of the list into two sublists, one with all the elements less than the pivot and another one with all the elements not less than the pivot. Recursively sort the sublists. Combine two sublists and pivot into final sorted list.

### **Requirement:**

The solution must follow the style of function merge sort on page 114. The help functions must be made inside let.

### **Sample Run:**

Please see the screen shot below to see the sample run



```
Project — run.x86-darwin — 80x24
Last login: Sat Jan  2 15:13:23 on ttys000
MACS-H-ZENG-MBA:~ h_zeng$ cd Documents/CSCI461VA/Project
MACS-H-ZENG-MBA:Project h_zeng$ sml
Standard ML of New Jersey v110.79 [built: Sun Oct  4 14:45:06 2015]
- use "CSCI461ProjTwo.sml";
[opening CSCI461ProjTwo.sml]
val partition = fn : int * int list -> int list * int list
val quickSort = fn : int list -> int list
val newQuickSort = fn : int list -> int list
val it = () : unit
- val x = [3, 1, 4, 8, 9, 5, 7, 6, 2, 0];
val x = [3,1,4,8,9,5,7,6,2,0] : int list
- quickSort x;
val it = [0,1,2,3,4,5,6,7,8,9] : int list
- █
```

### **File to turn in:**

Save your function in file names `YourNameProjTwo.sml` and submit it via blackboard link.

### **Due Date:**

Will be announced on blackboard.