COE 301: Computer Organization

MIPS Programming Assignment 2, Term 221

Quick Sort Recursive Function

The function **quick_sort** sorts an array of integers recursively. Translate this function into MIPS code. Write a **read_array** function that asks the user to input the number of elements **n** (must be greater than 1), allocates an array of **n** integers dynamically on the heap, reads **n** integers and stores them in the array, and returns the address of the allocated array and the number of elements in **$v0** and **$v1**, respectively. Write a function **print_array** that prints an array of **n** integers. Write a **main** function that calls the functions **read_array**, **print_array** (before sorting), **quick_sort**, and **print_array** (again after sorting). Test your program by entering different arrays and check if sorting works properly.

```
void quick_sort(int array[], int n) {
  int i = 0;                          // i = low index
  int j = n-1;                        // j = high index
  int pivot = array[(i+j)/2];         // pivot = middle value
  while (i <= j) {
    while (array[i] < pivot) i++;
    while (array[j] > pivot) j--;
    if (i <= j) {
      int temp = array[i];
      array[i] = array[j];            // swap array[i]
      array[j] = temp;                // with array[j]
      i++;
      j--;
    }
  }
  if (j > 0)   quick_sort(&array[0], j+1);       // Recursive call 1
  if (i < n-1) quick_sort(&array[i], n-i);       // Recursive call 2
}
```

```
(int array[], int n) read_array () {
  // Ask the user to enter n and allocate the array dynamically
  // Ask the user to input n elements of type int
  // The user input should be stored in array[]
  // Return the address of the array[] in $v0 and return n in $v1
}
void print_array (int array[], int n) {
  // Display the n elements of array[]
}
```

Submission Guidelines:

This assignment can be solved individually or in groups of two students only. No group should have more than two students. **At the beginning of your program, write the names of the students who worked on the program.** If this program was solved individually then write your name only. Your code should be well written and well document.

All submissions should be done through Blackboard. Submit the source code of the program. **If the assignment was solved by two students, then it is sufficient for one student to submit the assignment. The other student can write a note on Blackboard indicating his partner.**

All submissions should be done on time. Late submissions are accepted with a late penalty. The late penalty is **-1 point** for two days late. Blackboard will not accept any submission which is later than two days.

Grading Scheme:

Main function that calls *read_array*, *print_array*, and *quick_sort*	**[10 points]**
Allocating an array dynamically and reading *n* array elements	**[20 points]**
Sorting the array properly using recursive quicksort	**[40 points]**
Printing the array before and after sorting	**[20 points]**
Dividing the program into procedures and passing parameters properly	**[5 point]**
Program readability and comments	**[5 point]**
Total	**[100 points]**