COE 301 / ICS 233 – Computer Organization

MIPS Programming Assignment 2, Term 172

Due date: Sunday 11/03/2018 at 11:59 PM (Late submission = 0 points)

The following high-level language procedure recursively performs a binary search for a key within an array of sorted (ascending) integers. Translate this procedure into MIPS code. Write a main procedure to dynamically allocate an array of **n** integers (**n** is a user input that must be greater than 1), read an array of **n** sorted (ascending) integers, print the array of **n** sorted integers, print the index of the matching element if the key is found, otherwise print the value –(index where the key can be inserted). You need to write procedures for reading and printing the array, and pass parameters properly according to the MIPS convention.

```
// procedure:       Search sortedArray[first]..sortedArray[last] for key.
// returns:         index of the matching element if it finds key,
//                  otherwise  - (index where it could be inserted).
// parameters:      sortedArray: array of sorted (ascending) integers
//                  first, last: lower and upper subscript bounds.
// key:             integer to search for.
int BinarySearch(int sortedArray[], int first, int last, int key) {
   if (first <= last) {
       int mid = (first + last) / 2; // compute mid point.
       if (key == sortedArray[mid])
           return mid;
       else if (key < sortedArray[mid]) // Call BinarySearch for the lower part of array
           return BinarySearch(sortedArray, first, mid-1, key);
       else                             // Call BinarySearch for the upper part of array
           return BinarySearch(sortedArray, mid+1, last, key);
   }
   return -first;    // failed to find key
}

void read_array (int n, int sortedArray[]) {
   // Ask the user to input n sorted integer elements, and store them in sortedArray[]
}

void print_array (int n, int sortedArray[]) {
   // Display the n elements of sortedArray[]
}
```

Submission Guidelines:

All submissions should be done through Blackboard. Submit the source code of the program. Make sure that your program is well written and documented. The program will be graded according to its correctness and documentation. It is your responsibility to make sure that the program works. A program that does not assemble or run will receive zero on correctness. **Copying programming assignment is not allowed. This is individual work. Detected copies will get zero grades. This includes the one who wrote the program and the one who copied it.**

Grading Scheme:

Dividing the program into procedures and passing parameters properly	**[2 points]**
Dynamically allocating an array of *n* integers	**[3 points]**
Reading the sorted array integers and saving them in the allocated array	**[2 points]**
Properly search the array **recursively**	**[5 points]**
Printing the array after the search as well as the result of the search	**[2 points]**
Program readability and comments	**[1 point]**
Total	**[15 points]**