

CS2640 Project 3

Total points: 100

Project Specifications:

Write a program in MIPS assembly language that performs the following tasks:

1. Prompt the user for the quantity of signed integers that the user would like to enter.
2. Takes those integers as input from the console
3. With each integer entered, a subroutine must be called to store that integer into an array in ascending order. This subroutine will insert the integer into the array in the proper position. **DO NOT USE A SORTING FUNCTION.**
4. Print the sorted integers space-separated.
5. Prompt the user for a value for which to search the array.
6. A **recursive** subroutine must be called that performs a binary search for the value and should return true (1) if found, or, false (0) if not found..
7. The program must inform the user if the value was located or not.
8. The program should prompt the user for another search value.
9. The program should continue to prompt for input until the user terminates the program (using the stop button in Mars).

Be sure to use meaningful prompts when interacting with the user. Use as many subroutines as you would like to accomplish these tasks but you must implement the requirements above.

Be sure to Include pre and post condition comments for your subroutine

You should run your program choosing to enter 20 values.

- Enter unsorted integers
- Search for:
 - the minimum integer entered
 - the maximum integer entered
 - the median integer entered
 - an integer that doesn't exist in the array
- Save a copy of this test; **no screenshots, copy and paste the console to a text file.**

Submission Instruction: You must run the assembly program using the SPIM simulator. Please zip the following files:

- The assembly program (using file extension .asm) and the console output text file for the test stated above..
- Copy your files to a folder named, "your_name_p3" where "your_name" is your first and last name separated by an underscore. Compress that folder into a single Zip archive which should be named, "your_name_p3.zip"

Submit your zip file via Blackboard

A recursive binary search algorithm:

```
binarySearch(&array, start, end, searchVal){
    if (start > end)
        return false

    mid = start + (end - start)/2;

    if (array[mid] == searchVal)
        return true;

    if (array[mid] > searchVal)
        return binarySearch(array, start, mid-1, searchVal);

    return binarySearch(array, mid+1, end, searchVal);
}
```