Lab #3: Integer Sort in RISC-V Assembly

Introduction

In this project you are asked to implement counting sort using RISC-V assembly code. In particular, consider the following pseudocode.

```
int   keys[numkeys];
int   output[numkeys];

countingsort(int *keys, int *output, numkeys, maxnumber) {
        int   count[maxnumber+1], n;
        for (n = 0; n++; n ≤ maxnumber)
                count[n] = 0;
        for (n = 0; n++; n < numkeys)
                count[keys[n]]++;
        for (n = 1; n++; n ≤ maxnumber)
                count[n] = count[n] + count[n-1];
        for (n = 0; n++; n < numkeys) {
                output[count[keys[n]]-1] = keys[n];
                count[keys[n]]--;
        }
}
```

Here ***keys*** is the array of integer items of length ***numkeys*** to be sorted with maximum integer ***maxnumber***, while ***output*** is the sorted array.

Project Tasks

Your code should be written as a function, which takes pointers to the ***key*** and ***output*** arrays in registers ***$a0*** and ***$a1***, and ***numkeys*** and ***maxnumber*** in registers ***$a2*** and ***$a3***, correspondingly. Use skeleton sort.s to write your code. Note that sort.s has an example of an array to sort, however your code will be checked against other arrays so make sure that it works correctly with different inputs.

Lab Report

Your lab report should be a single PDF file, which has all the following items in the following order and clearly labeled:

1. **Please indicate how many hours you spent on this lab.** This will be helpful for calibrating the workload for next time the course is taught.

2. In case you modify the original code (e.g. changing for-loop to do-while loop) show modified C code in your report that is closest to the implemented assembly
3. Your "sort.s" file *
4. If you have any feedback on how we might make the lab even better for next quarter, that's always welcome. Please submit it in writing at the end of your lab

* Footnotes:

   a. For your code files, please copy and paste your code clearly in a monospace font (ex., Courier New).
   b. In addition to part 2, provide in the comment the equivalent C code for each added line of the code in the skeleton.

RTL Autograder

You will need to submit your "sort.s" file to the Gradescope assignment, "test lab3". When you make a submission, an autograder will run several tests to ensure your design is correct. You must pass all the tests to get a full score on the lab. You can resubmit as many times as you like.

What to Turn In

You will need to make two separate Gradescope submissions: "lab3 Code" and "lab3 Report".

- For "lab3 Code", please submit your "sort.s" files.
- For "lab3 Report", please submit your lab report PDF file.

Only one submission per group is needed; be sure to add all your group members.