

Zach Dubinsky

CSC-270

Prof. Rieffel

2/15/22

Lab 6

I affirm that all included work is my own in accordance with the class syllabus and the Union College Honor Code [Signed: Zachary Dubinsky, 2/15/22].

1. A Remark

I did not use the stack very much because I didn't want to preserve registers that do not need preserving. I recall that the `$v_`, `$a_`, and `$t_` registers are not preserved so I made no effort to do so. The program works fully but I can see how this might be problematic going forward.

2. Array-Sum.asm

My array sum algorithm iterates over each element of the array, using a counter, and adding the value of the element at each index into the `$v0` return register. Before entering the loop a condition is checked to ensure that the array is not empty, in which case the subroutine ends. Following this check, there is a jump to a simple for loop that sums the elements of the array, jumping to the end of the subroutine when the index counter is greater than the number of elements in the array. The following console output shows the result of the subroutine on the array `[100,2,-395,2993,2,-400,203]`.

```
Console
Sum: 2505
```

3. Insertion-sort.asm

My insertion sort algorithm uses the pointer arithmetic algorithm. The following images illustrate a successful sort. The first is the console output, the second is the data section of QtSpim before the sort, and the third is the data section of QtSpim following the sort. A remark on the algorithm, the outline asks that a parameter be provided with the number of elements in the array to the insertion sort subroutine. This is trivial for pointer arithmetic, so I provided it and used it to calculate the end address of the array, which is also trivial since the end address is needed to calculate the number of elements.

```
Console
Unsorted Array: -1,0,5,-9,12,13,1,4,-5,2,8,8891,-2
Sorted Array: -9,-5,-2,-1,0,1,2,4,5,8,12,13,8891
```

```
User data segment [10000000]..[10040000]
[10000000]..[1000ffff] 00000000
[10010000] 1869835861 1684370546 1920090400 0540703073 Unsorted Array:
[10010010] 1919898368 0543450484 1634890305 0002112121 .Sorted Array: .
[10010020] 0000655404 -1 0000000000 0000000005 .
[10010030] -9 0000000012 0000000013 0000000001 .
[10010040] 0000000004 -5 0000000002 0000000008 .
[10010050] 0000008891 -2 0000000000 0000000000 .
[10010060]..[1003ffff] 00000000
```

```
User data segment [10000000]..[10040000]
[10000000]..[1000ffff] 00000000
[10010000] 1869835861 1684370546 1920090400 0540703073
[10010010] 1919898368 0543450484 1634890305 0002112121
[10010020] 0000655404 -9 -5 -2
[10010030] -1 0000000000 0000000001 0000000002
[10010040] 0000000004 0000000005 0000000008 0000000012
[10010050] 0000000013 0000008891 0000000000 0000000000
[10010060]..[1003ffff] 00000000
```