

CSC 362 Program #5

Due Date: Tuesday, November 11

Implement the Selection Sort algorithm for an array of int values in assembly language. The selection sort algorithm is given below:

Given int a[] and int n (the number of elements in the array), selection sort iterates for i from 0 to n-2, each time searching array locations i through n-1 for the smallest element and swapping that element with a[i]. For instance, if our array is:

7, 4, 2, 6, 3 → iteration i=0: search elements 0-4, the smallest is 2, swap 2 and a[0]

2, 4, 7, 6, 3 → iteration i=1: search elements 1-4, smallest is 3, swap 3 and a[1]

2, 3, 7, 6, 4 → iteration i=2: search elements 2-4, smallest is 4, swap 4 and a[2]

2, 3, 4, 6, 7 → iteration i=3: search elements 3-4, smallest is 6, swap 6 and a[3]

Done

The algorithm:

```
for(i=0;i<n-1;i++)
{
    min = a[i];
    minposition = i;
    for(j=i+1;j<n;j++)
        if(a[j] < a[minposition])
        {
            min = a[j];
            minposition = j;
        }
    swap a[i] and a[minposition]
}
```

Implement your code in assembly language. Use C code to initialize any variables and perform output only. All other code must be handled in assembly language. Note that you may find this program tricky because you are limited to 4 data registers. Your program will be organized as follows: initialize the int array and the number of elements of the array in C code (hardcode the array rather than inputting it). Enter assembly code to perform the sort. All logic for the sorting algorithm must be done in assembly code. Once sorted, exit assembly code and output the sorted array in C code (using a for loop). Use only one array (do not manipulate a copy of the array). Run the program on the 2 data sets below. Remember to comment your code, especially with respect to how it helps accomplish sorting.

Run #1: 100, 99, 97, 95, 90, 87, 86, 83, 81, 77, 74, 69, 63, 50, 44, 43, 39, 31, 29, 12

Run #2: 8, -1, 7, -8, 6, 3, -4, 4, 1, -2, 9, -5, 0, 2

Hand in your program and the output for both runs.