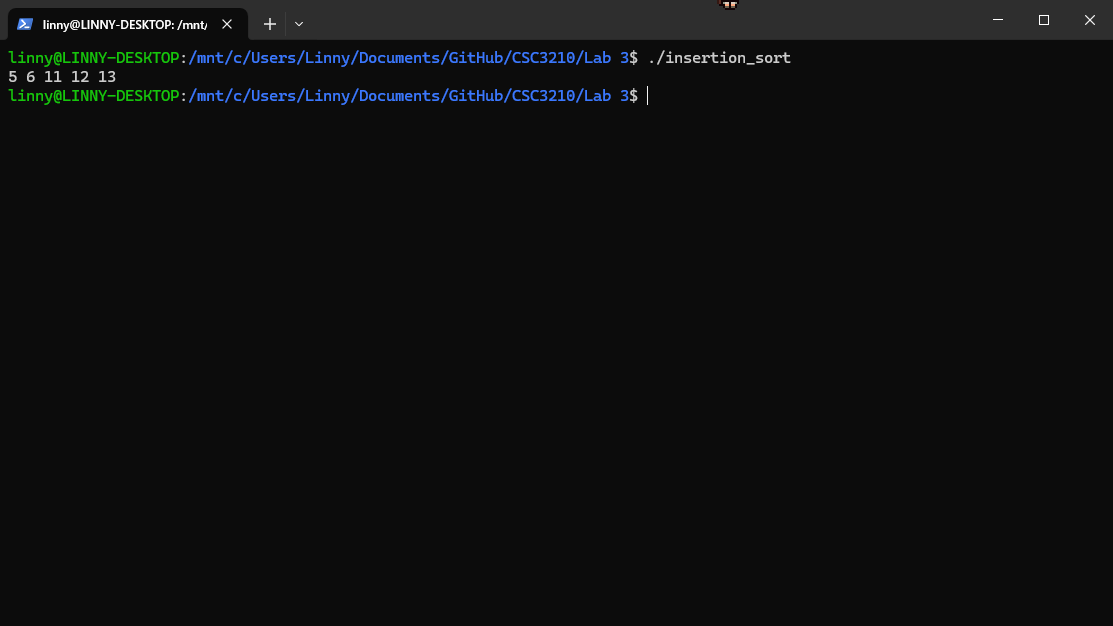
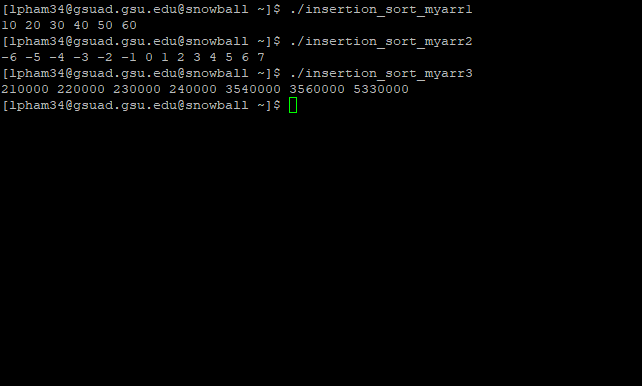
**Name:   
Panther ID:  
  
Task 1 Insertion Sort (40pts)**

1. (5 pts)Download the insertion\_sort.c code from iCollege and compile the binary and assembly of this program using gcc.

*Insert screenshots of compilation*



1. (20 pts)Convert the above code, addTwo.s to RISC-V, and run the following test cases:  
   myarr1 = [10, 20, 30, 40, 50, 60]  
   myarr2 = [7, 6, 5, 4, 3, 2, 1, 0, -1, -2, -3, -4, -5, -6]  
   myarr3 = [240000, 3560000, 230000, 3540000, 220000, 5330000, 210000]  
   *Insert screenshots of the task*  
   
2. (15 pts)How many registers does it take to store each array for each case in B?  
     
     
   myarr1: 6 registers

myarr2: 14 registers

myarr3: 7 registers

**Task 2 Tower of Hanoi, with Recursion (60pts)**

1. (30pts)Describe how to implement the ToH recursive solution in assembly in your own words. You can use diagrams to show what registers and operations to use.

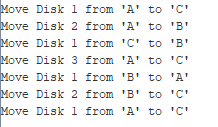
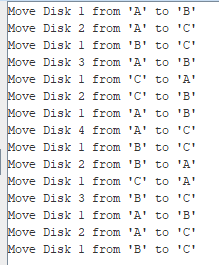
Set up the stack frame and register usage

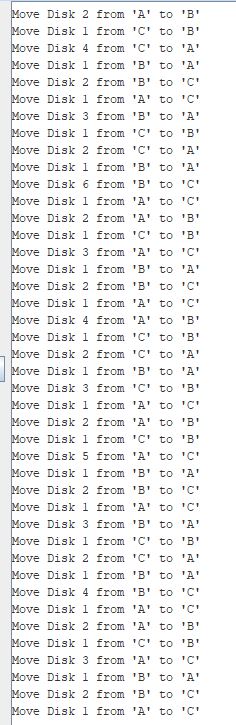
Implement the base case

Implement the recursive case:

* Decrement the number of disks
* Make recursive call to move n-1 disks from source to auxiliary
* Print move of largest disk from source to destination
* Make recursive call to move n-1 disks from auxiliary to destination

Return

1. (30pts)Implement the ToH recursive solution in RISC-V assembly and demo cases with 3,4,and 7 disks.  
     
     
   3   
   4

7 

1. (10pts)What is the largest number of disks you can solve with your code? Justify your answer.  
   The max value is 2,147,483,647, so theoretically the program can handle up to 31 disk because the program has time complexity of O(2n)