

# CS 2110

## Timed Lab 2

### Due Date and Time

Day: Thursday, June 30<sup>th</sup>, 2015

Time: 5:45pm

## Policy

### Submission

**TURN IN THIS ASSIGNMENT ELECTRONICALLY USING T-SQUARE. SUBMISSIONS WHICH ARE LATE WILL NOT BE ACCEPTED. EMAIL SUBMISSIONS WILL NOT BE ACCEPTED UNDER ANY CIRCUMSTANCES! IN ADDITION IF YOU FORGET TO HIT THE SUBMIT BUTTON YOU WILL GET A ZERO.**

### Questions

If you are unsure of what questions mean, the TA's will clarify them to the best of their ability. In the end you are solely responsible for what you submit. We will not be able to answer any questions about how to reach a solution to the lab questions.

### What's Allowed

- The assignment files
- Your previous Homework and Lab submissions
- Your mind
- Blank paper for scratch work

### What's Not Allowed

- The Internet (except the T-Square Assignment page to submit)
- Any resource on T-Square that is not given in the assignment.
- Dropbox (if your harddrive crashes we will let you retake it!)
- Notes on paper or saved on your computer.
- Textbook
- Email
- IM
- Contact in any form with any other person besides TA's
- If you have any questions on what you may not use then assume you can't use it and ask a TA.

## **Other Restrictions**

1. You may not leave the classroom until we have verified that you have submitted the lab. If you leave the classroom without submitting you will receive a zero.
2. **YOU MUST SUBMIT BY THE END OF YOUR LAB PERIOD.** Bear in mind that the clock on your computer may be a few minutes slow. You are supposed to have a full class period to work, and we are letting you use the 10 minutes between classes to make sure you have submitted your work. **WE WILL NOT ACCEPT LATE SUBMISSIONS**, be they 1 second or 1 hour late.
3. The timed lab has been configured to accept one submission. If you accidentally submit or submit the wrong version, call one of the TA's and we will reopen submission for you. But PLEASE PLEASE PLEASE submit the right thing the first time. The TA's get busy at the end of the lab making sure everyone submitted, and it's tough doing that AND re-opening submissions for 5 students. Yes, it does happen. Don't let it happen to you.

## **Violations**

Failure to follow these rules will be in violation of the Georgia Tech Honor Code. **AND YOU WILL RECIEVE A ZERO** and you will be reported to the professor and the Office of Student Integrity. We take cheating and using of unauthorized resources **VERY SERIOUSLY** and you will be in serious trouble if you are caught.

## **Remember**

1. Please don't get stressed out during a timed lab. You have plenty of time however use your time effectively
2. Partial credit is given. If you don't know something at least **TRY** do not just walk out of the lab or submit an empty file. Do the best you can!
3. Make sure your code can compile. Your code must compile to get any points for this assignment!
4. Remember what you can and can't use if you don't know then don't use it and ask a TA if you can use it. If we catch you with unauthorized resources we will give you a zero, so better to be safe than sorry.

## Hanoi

The Tower of Hanoi is a classic puzzle game that uses  $n$  disks of distinct sizes and three rods. Initially, the disks are stacked in ascending order of size on rod A. The objective is to move all disks to rod C while following these three rules:

1. Only one disk can be moved at a time.
2. Only the top disk can be moved from one rod to another.
3. A smaller disk cannot be placed below a larger disk.

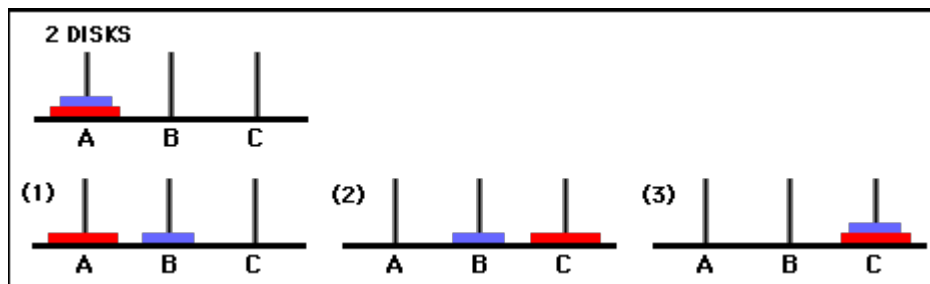
You will be writing a recursive assembly program which will solve this puzzle using the least possible number of moves. Your code **MUST** follow the LC-3 Calling Convention. Use the C code below and use the template of hanoi.asm for this part. Also, when you are printing out the move, make sure the follow the format as specified in the C code. For example, calling `hanoi('A', 'C', 'B', 2)` should print:

A->B

A->C

B->C

Lastly, store the number of moves into the label named ANSWER in hanoi.asm. Here is an image to help visualize what the above test case does.



```
int hanoi(char* start, char* end, char* aux, int n) {
    int moves = 0;
    if(n<1)
        return 0;
    if(n<=1) {
        printf("%c->%c\n",start,end);
        return 1;
    }
    moves = moves + hanoi(start, aux, end,n-1);
    printf("%c->%c\n",start,end);
    moves = moves + 1;
    moves = moves + hanoi(aux, end, start, n-1);
    return moves;
}
```

Tips:

1. The `char*` means to pass in the ADDRESS of the character! Not the character itself. DO THIS. Points will be deducted if you do not do this.
2. Don't worry if your algorithm does not work for large numbers. This algorithm runs in  $O(2^n)$ .
3. FOLLOWING THIS PSUEDO CODE IS IN YOUR BEST INTEREST

## Deliverables:

- hanoi.asm