

frankSJSU DataStructure

 Search this site

Frank's Home Page

[CMPE126 home](#)[Greensheet](#)

Frank's Notes

[operator overloading](#)[storing objects](#)[pointer & deep copy](#)[array of objects](#)[linked list](#)[variable size objects](#)[create a linked node](#)[create a linked list](#)[linked list insertion](#)[find middle](#)[hybrid list](#)[linked list quiz](#)[recursion](#)[stack](#)[stack with array](#)[math expression](#)[queue](#)[simulation](#)[frankSimulation s16](#)[priority queue & heap](#)[search by hashing](#)

Frank's Slides

Frank's Code

Programming Exam

[PE #3 F16](#)[PE #1 guide F15](#)

Midterm Exams

[midterm 2 F19](#)[midterm 1 S18](#)[midterm 2 F17](#)[midterm 1 F17](#)[midterm 2 S17](#)[midterm1 S17](#)[midterm2 F16](#)[midterm1 F16](#)[Labs and Homeworks >](#)

Lab 5 Recursion

Objectives:

1. [Practice recursive functions in both array and linked list](#)

Overview

1. Go back to complex number in [Lab 2 where you store more than 15 complex numbers in an array.](#)
2. Find the largest complex number in the array using recursive method. The largest number is the one who has the largest $(\text{real}) * (\text{real}) + (\text{img}) * (\text{img})$
3. Go back to forward link list in [Lab 3 where you store many stocks in a link list.](#)
4. Write the reverse print function using recursive method.

Discussions

1. The textbook has very detail implementation of both routines. You're encouraged to reference it.
2. In order to easily compare complex numbers, do you think overloading a less than ($<$) or greater than ($>$) operator will make code cleaner? The example used in [linked list insertion page](#) has similar operator overloading to facilitate in-order insertion.
3. If you want more, do the tower of Hanoi and count the number of recursions (you can validate your result with [wikipedia's number](#)).

Comments

You do not have permission to add comments.

midterm S16

Final Exams

Final S17

Final S16

Final F15

Final S15

Labs and Homeworks

Misc Lab FYI

Lab 0 C++

Lab 1 classes

Lab 2 object array

Lab 3 Linked List

Lab 4 Doubly Linked List

Lab 5 Recursion

Lab 6 Stack

Lab 6+ math expression

Lab 7 Simulation

Lab 7a Palindrome

Lab 8 search

Lab 9 hashing

Lab 10 sort

[Sign in](#) | [Recent Site Activity](#) | [Report Abuse](#) | [Print Page](#) | Powered By [Google Sites](#)