

# C Programming: Pointers recap, pass-by-value vs. pass-by-reference, stack vs. heap

Yipeng Huang

Rutgers University

February 2, 2021

# Table of contents

## Announcements

Quiz class feedback

Programming assignment

`pointers.c`: A lab exercise for pointers, arrays, and memory

Lesson 6: Arrays are just places in memory

Lesson 7: Passing-by-value

Lesson 8: Passing-by-reference

Lesson 9: Passing an array leads to passing-by-reference

Lesson 10: How the stack works; recursion example

`dotProduct.c`: A program to calculate the dot product of two vectors

# Quiz class feedback

## Connecting to ilab, command line, transferring files

- ▶ Check out the TAs' recitation slides on Canvas, Files section.
- ▶ `https://resources.cs.rutgers.edu/docs/new-users/beginners-info/`
- ▶ Research, practice, and improve as you go.

## Programming in C: Syntax and debugging

- ▶ `https://resources.cs.rutgers.edu/docs/new-users/beginners-info/`
- ▶ "Beginners info" is a misnomer: that page is useful for everyone.
- ▶ Study and understand the examples given in class, in the 2021\_0s\_211 repository, and in Modern C book Chapters 1–8.
- ▶ `http://www.cs.yale.edu/homes/aspnes/pinewiki/C(2f)Debugging.html`

# Quiz class feedback

## Handling files and working with strings in C

- ▶ [https://www.tutorialspoint.com/c\\_standard\\_library/c\\_function\\_fscanf.htm](https://www.tutorialspoint.com/c_standard_library/c_function_fscanf.htm)
- ▶ Example files in the 2021\_0s\_211 repository.

## Memory, pointers, malloc and free

# Programming assignment

## Programming assignment

- ▶ Due in 9 days: 11:59pm Thursday, February 11.
- ▶ Find class's frequently asked questions on Piazza.
- ▶ As of last week, you should have everything you need for at least part 1, goldbach, part 2, maximum, and part 3, matMul.
- ▶ Goal for today, Tuesday: Solidify our discussion about pointers. An example of an assertion.
- ▶ Goal for Thursday: Work through an example of building a queue using structs and pointers. Everything you need for part 4, balanced, and part 5, bstReverseOrder. Debugging, GDB?

# Table of contents

## Announcements

Quiz class feedback

Programming assignment

`pointers.c`: A lab exercise for pointers, arrays, and memory

Lesson 6: Arrays are just places in memory

Lesson 7: Passing-by-value

Lesson 8: Passing-by-reference

Lesson 9: Passing an array leads to passing-by-reference

Lesson 10: How the stack works; recursion example

`dotProduct.c`: A program to calculate the dot product of two vectors

# Why pointers?

Pointers underlie almost every programming language feature:

- ▶ arrays
- ▶ pass-by-reference
- ▶ data structures

Vital reason why C is a low-level, high-performance, systems-oriented programming language (why we use it for this class, computer architecture).

# git pull

- ▶ From the folder `2021_0s_211`, type: `git pull`.
- ▶ By now we have several example codes: `isPrime.c`, `numList.c`, `pointers.c`, `dotProduct.c`.
- ▶ This hands-on-lab is in `pointers.c`.



## Lesson 6: Arrays are just places in memory

- ▶ name of array points to first element
- ▶ `malloc()` and `free()`
- ▶ stack and heap
- ▶ using pointers instead of arrays
- ▶ pointer arithmetic
- ▶ `char* argv[]` and `char** argv` are the same thing

## Lesson 7: Passing-by-value

Using stack and heap picture, understand how pass by value and pass by reference are different.

- ▶ C functions are entirely pass-by-value
- ▶ `swap_pass_by_values()` doesn't actually succeed in swapping two variables.

## Lesson 8: Passing-by-reference

Using stack and heap picture, understand how pass by value and pass by reference are different.

- ▶ You can create the illusion of pass-by-reference by passing pointers
- ▶ `swap_pass_by_references()` does succeed in swapping two variables.

## Lesson 9: Passing an array leads to passing-by-reference

## Lesson 10: How the stack works; recursion example

Low addresses		Global / static data
	Heap grows downward	Dynamic memory allocation
High addresses	Stack grows upward	Local variables, parameters

[Table: Memory structure](#)

# Table of contents

## Announcements

Quiz class feedback

Programming assignment

`pointers.c`: A lab exercise for pointers, arrays, and memory

Lesson 6: Arrays are just places in memory

Lesson 7: Passing-by-value

Lesson 8: Passing-by-reference

Lesson 9: Passing an array leads to passing-by-reference

Lesson 10: How the stack works; recursion example

`dotProduct.c`: A program to calculate the dot product of two vectors

# dotProduct.c: A program to calculate the dot product of two vectors

## Examples of:

- ▶ `fscanf()` for integers, floating point, and strings.
- ▶ Arrays, `malloc` and `free`, and pointer arithmetic.
- ▶ Assertions.

Study how we deliberately work with vectors `a` and `b` using different syntax to show different approaches.