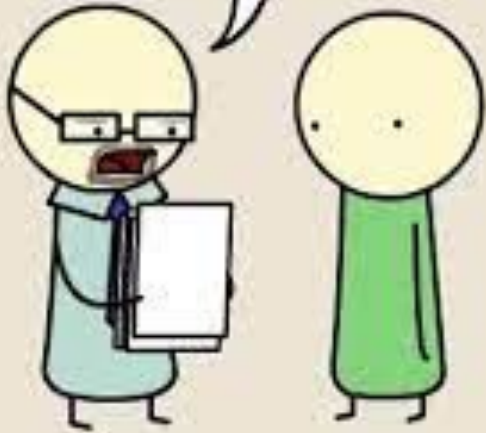


C

THIS IS GREAT, BUT YOU FORGOT TO ADD
A NULL TERMINATOR. NOW I'M JUST READING
GARBAGE.



Section 5 Chars and Strings

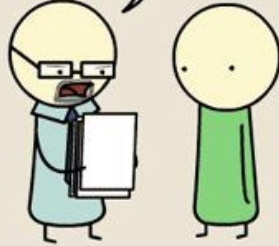
CS 107A, Autumn 2021
Andrew Benson (adbenson@)



Don't forget to start recording

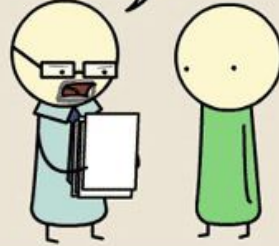
PYTHON

THIS IS PLAGIARISM.
YOU CAN'T JUST "IMPORT ESSAY."



JAVA

I'M TWO PAGES IN AND I STILL
HAVE NO IDEA WHAT YOU'RE SAYING.



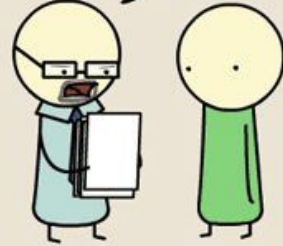
C++

I ASKED FOR ONE COPY,
NOT FOUR HUNDRED.



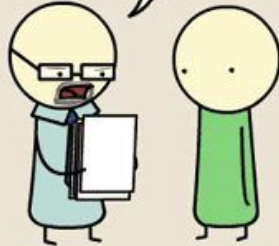
UNIX SHELL

I DON'T HAVE PERMISSION TO
READ THIS.



ASSEMBLY

DID YOU REALLY HAVE TO REDEFINE EVERY
WORD IN THE ENGLISH LANGUAGE?



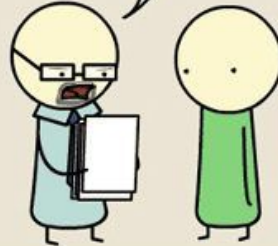
C

THIS IS GREAT, BUT YOU FORGOT TO ADD
A NULL TERMINATOR. NOW I'M JUST READING
GARBAGE.



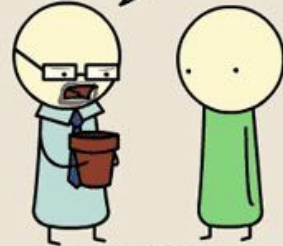
LATEX

YOUR PAPER MAKES NO GODDAMN SENSE,
BUT IT'S THE MOST BEAUTIFUL THING
I HAVE EVER LAD EYES ON.



HTML

THIS IS A FLOWER POT.





Unix Tip Spotlight

- How do you clear the screen?
 - `clear`
 - Does what it says on the tin
 - `<CTRL+L>`
 - Does the same thing as `clear`
 - Doesn't involve an actual command, so it's more...aesthetic?
 - Neither of these necessarily deletes the screen output – your terminal probably allows you to scroll up to see the cleared output!



Announcements

- Website Updates
 - Slight schedule change
 - Gradebook
 - Let me know if anything seems wrong
- Pls enroll, you can always drop later
 - Add/drop deadline is Friday
- 1:1s (Thursday, Friday, Monday):
<https://calendly.com/adbenson/cs107a-1-1>



Bird's Eye View

Day	Week 3 Monday	Tuesday	Wednesday	Thursday	Friday	Week 4 Monday	Tuesday	Wednesday
CS 107A		Section: Chars and Strings		Section: More C-Strings			Section: Pointers and Memory	
CS 107	Lecture: Chars and C Strings		Lab 2: C-Strings		Lecture: Arrays and Pointers	Lecture: Stack and Heap		Lab 3: Arrays / Pointers
CS 107 assignments			assign1 due, assign2 released					assign2 due, assign3 released



Agenda

- Chars
- Strings
- String Functions
- Practice: `filter_pollution`
- Practice: `string_reverse` and `is_string_palindrome`



Chars



Chars

- 1 value, 2 interchangeable interpretations
- Interpretation 1: 8-bit integer value
- Interpretation 2: A particular character in the ASCII table
 - `man ascii` shows you an ASCII table
- `'e' - 'a' == 4 // true!`



Important Chars

- The NUL character `'\0'` (value: 0)
- The newline character `'\n'` (value: 10)
- The space character `' '` (value: 32)
- The digit `'0'` (value: 48)
- Uppercase `'A'` (value: 65)
- Lowercase `'a'` (value: 97)



Strings

Strings [of characters]

- Strings are just contiguous sequences (“arrays”) of characters in memory
- Valid strings need to be terminated by a NUL character
 - Without it, they’re still char arrays, just not strings
- Num bytes used = string length + 1

"Hello"

index

0

1

2

3

4

5

char

'H'

'e'

'l'

'l'

'o'

'\0'



Strings Creation

- String literals
 - Read-only
 - Whenever a literal string in double quotes is assigned to something other than a char array
 - `char *literal_string = "I am a literal string";`
- Stack strings
 - Modifiable array on the stack, only valid within that stack frame
 - `char buffer[20]; buffer[0] = '\0';`
 - `char buff[6]; strcpy(buff, "hello");`
 - `char buf[] = {'a', 'b', 'c', '\0'};`
 - `char b[] = "abc";`



Strings

- What does "" look like in memory?
- What does "hello" look like in memory?



String Functions



I want to do X to a valid C string:

Access (read/write) the ith character	Use array indexing
Get the length of the string	<code>strlen</code>
Check whether two strings are equal	<code>strcmp</code> , <code>strncmp</code>
Compare two strings lexicographically	<code>strcmp</code> , <code>strncmp</code>
Copy a string into an already existing buffer	<code>strcpy</code> , <code>strncpy</code>



I want to do X to a valid C string:

Find the location of a specific character	<code>strchr</code>
Find the location of a specific substring	<code>strstr</code>
Append a string to another, assuming the latter has extra room	<code>strcat, strncat</code>
Split a string based on delimiters	<code>strtok</code>
Parse a string based on accept/reject lists	<code>strspn, strcspn</code>



Section 5 Worksheet

```
git clone /afs/ir/class/cs107a/WWW/git/section5
```

- Practice: `filter_pollution`
- Practice: `string_reverse` and `is_string_palindrome`