

# CSE 374: Programming Concepts and Tools

Eric Mullen

Spring 2017

Lecture 11: strings, aliasing

# Administrivia

- HW1: 'uname -v' is ok, we'll give you points back for it, sorry for inconsistency in grading
- HW3: due tomorrow night!!! It takes some time
- Friday is Engineering Discovery Days. Go if you must. We'll still have class.
- Midterm next Fri (4/28), Homework due night before
- Review Session on Tues from 4-6pm in CSE 203

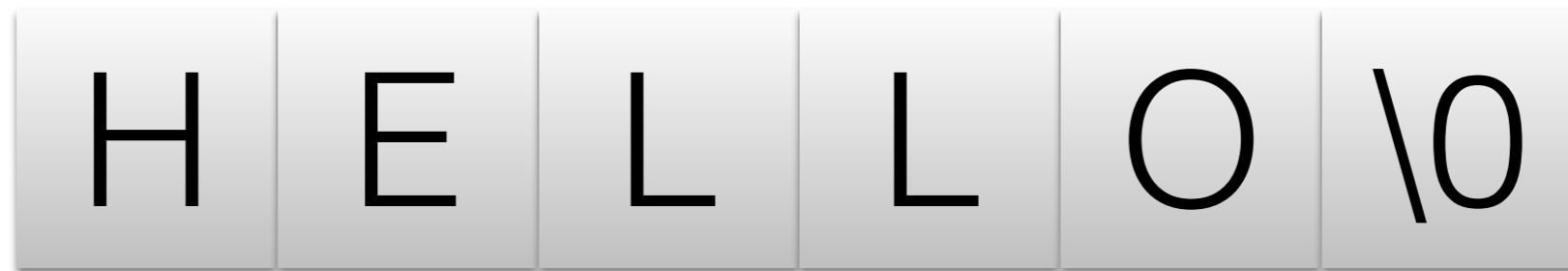
# Where we are

- C does a lot of things, despite being a small language
- Pointers are extremely useful, powerful, and dangerous
- Today:
  - Strings in C
  - Aliasing
  - gdb

# Strings

- C doesn't really have strings
- `char*` usually means string
- All code *must agree* on how to treat these strings
  - Pay attention: many of the errors I see in this class are issues with strings

# Strings



- Strings don't come with a length
- Strings are instead null terminated (end with a '\0' byte)
- Lots of functions in `<string.h>`

# String Functions

- `strlen`: get length of a string
- `strcpy`: string assignment (copy one string to another)
- `strncpy`: like `strcpy`, but safer (provide additional bound)
- `strcat/strncat`: string append
- `strchr`: find first occurrence of byte in string
- `strstr`: find first occurrence of substring in string

# Aliasing

```
void f() {  
    int i=17;  
    int x=3;  
    int j=g(&i, &x);  
    printf("%d %d %d", i, j, x);  
}
```

i	18
x	4

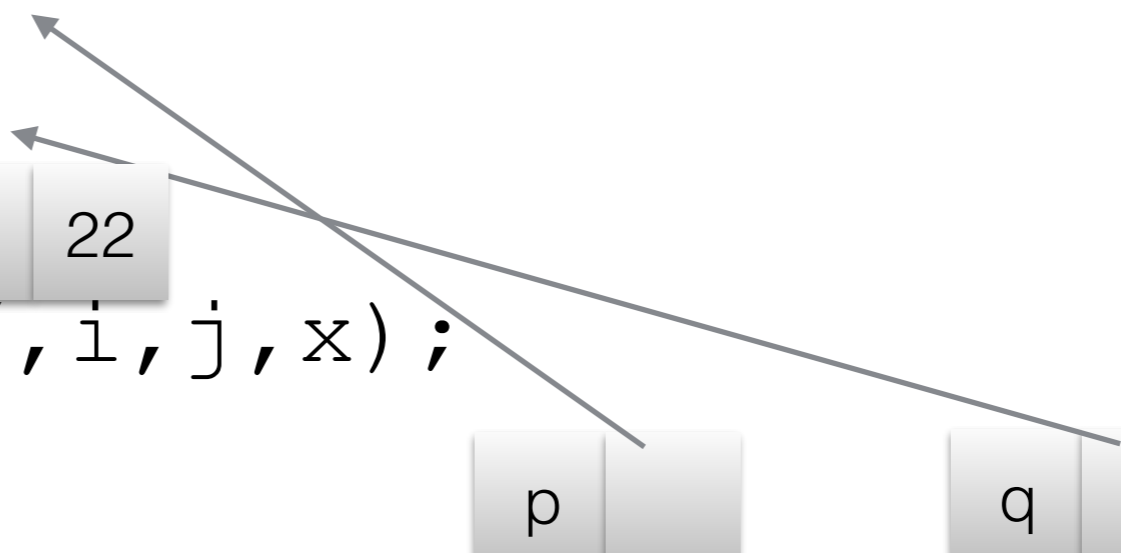
j	22
---	----

p	
---	--

q	
---	--

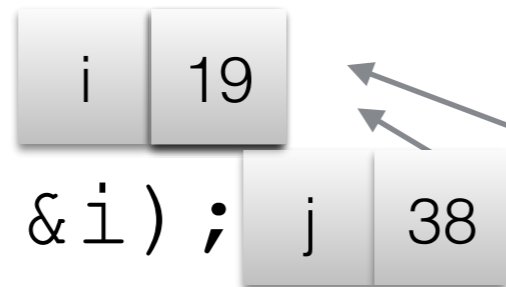
```
int g(int *p, int* q) {  
    *p = (*p) + 1;  
    *q = (*q) + 1;  
    return (*p) + (*q);  
}
```

22
----



# Aliasing

```
void f() {  
    int i=17;  
    int j=g(&i, &i);  
    printf("%d %d", i, j);  
}
```



```
int g(int *p, int* q) {  
    *p = (*p) + 1;  
    *q = (*q) + 1;  
    return (*p) + (*q);  
}
```

