CSc 352

Out-params

Benjamin Dicken

More than one value?

- In C, you can return one value from a function (pointer, int, char, etc)

- What if you want to return more than one value?

- For example, a function that:
  - Splits a C string exactly in half, and returns both halves
  - Takes a physical address, returns a lat and long value
  - . . . .

Out-Parameters

- An out-parameter is a way of getting a value “out” of a function call without relying on a **return** statement
- If you are calling function Y from function X, you can send Y the address of a local variable from X to store a value into
- This gives the ability to “return” multiple things!

```c
void split_in_half(char* to_split, char** half_one, char** half_two) {
  int half = (int) (strlen(to_split) / 2);
  *half_one = calloc(sizeof(char), half+1);
  *half_two = calloc(sizeof(char), half+1);
  strncpy(*half_one, to_split, half);
  strncpy(*half_two, (to_split+half), half);
}

int main() {
  char alphabet[27] = "abcdefghijklmnopqrstuvwxyz";
  char * h1;
  char * h2;
  split_in_half(alphabet, &h1, &h2);
  printf("alphabet: %s\n", alphabet);
  printf("h1: %s\n", h1);
  printf("h2: %s\n", h2);
  return 0;
}
```

Activity

Implement the function

- Rewrite **dynamic_strcat** to return void, instead give resulting concatenated string back via an out-parameter
- Thus, function should have three total arguments (two “regular” arguments, and one out-param)