

# CS 61C Fall 2014 Discussion 1 – C

## Uncommented Code? Yuck!

The following functions work correctly (note, this does not mean intelligently), but have no comments. Document the code to prevent it from causing further confusion

Function 1:

```
int foo(int *arr, size_t n) {
    return n ? arr[0] + foo(arr + 1, n - 1) : 0;
}
```

Function 2:

```
int bar(int *arr, size_t n) {
    int sum = 0, i;
    for (i = n; i > 0; i--) {
        sum += !arr[i - 1];
    }
    return ~sum + 1;
}
```

Function 3:

```
void baz(int x, int y) {
    x = x ^ y;
    y = x ^ y;
    x = x ^ y;
}
```

## Programming With Pointers

Complete the implementation of the following functions based on the comments.

1. Swaps the value of two ints outside this function.
2. Increments the value of an int outside this function by one.
3. Returns the number of bytes within a string. Do not use strlen().

## Got a Problem?

Examine the following snippet of code and circle any errors you see. Then for each error, state how you would fix the problem in the space below each set of statements. The function `strlen()` returns the length of a null-terminated string, and the function `sprintf()` writes a null-terminated formatted string into a given character buffer.

```
/* #include statements are located above */
int random_num(int n); // generates a random number in range [0, n-1]
void alloc_failed(); // called when memory allocation fails

typedef struct song {
    char identifier; // format of string is "<title> by <artist>"
    int duration; // duration in seconds
} song_t;

void set_identifier(song_t* song, char* title, char* artist) {
    int len = strlen(title) + strlen(artist) + 4; // strlen(" by ") = 4
    song->identifier = (char *) malloc(len);
    if (!song->identifier) alloc_failed();
    sprintf(song->identifier, "%s by %s", title, artist);
}

song_t* create_song(char* title, char* artist, int duration) {
    song_t my_song;
    set_identifier(&my_song, title, artist);
    my_song.duration = duration;
    return my_song;
}

void edit_identifier(song_t* song, char* new_title, char* new_artist) {
    set_identifier(song, new_title, new_artist);
}

song_t* pick_random_long_song(song_t songs[], int min_duration) {
    int i, len, count;
    bool has_long_song = false;
    len = sizeof(songs);
    song_t* long_songs[len];
    for (i = 0; i < len; i++) {
        if (songs[i]->duration >= min_duration) {
            has_long_song = true;
            long_songs[count++] = songs[i];
        }
    }
    int index = random_num(count);
    return has_long_song ? long_songs[index] : NULL;
}
```