NWEN 241
Systems Programming

Week 3 Tutorial

`sscanf()` and `sprintf()` functions

- `scanf()` and `printf()` functions are used to read from and write to the standard input/output
- `sscanf()` and `sprintf()` are used for the same goal but instead of the standard input/output, **they use strings**
- One of their main advantage is when you need to prepare a string for later use

Structures

```
// declare "struct person"  type
struct person {
    char name[100];
    int age;
};

// give it an alias person_t
typedef struct person person_t;
```

- Struct is just a collection of variables (which can have different types) under a single name
- You can access members with the '.' operator or through a pointer with the '->' operator
- A struct can be referenced, copied, and assigned to
- The size of a struct is guaranteed to be as large as the sum as the size of its members

* And &

	In Declaration	In Expression
*	`int *i;` *Declare `i` as a pointer to `int`*	`*i` *Dereference `i` or obtain the value that `i` points to*
&	*N/A (available in C++)*	`&i` *Get the address of `i` (a pointer to `i`)*

Pointers and Arrays

- **Array decays into a pointer: an array is just a fixed pointer**
- You cannot re-assign an array to point to another location
- You can let another pointer point to the array

```
int *p;
```

- p can point to an `int`
- p can point to an array of `int`

Introducing GDB

- GDB: GNU Debugger

- A much better way to debug your programs
  - No need to rely on `printf()` to see the values of the variables
  - You can step through your code
  - You can even change variable values!!!

- You will learn more about GDB in Exercise 2 (out on Tue, 22 March)