Use `sscanf` to Read Formatted Input from a String

To read formatted input from a string:

```
int sscanf (const char* s,
    const char* format, ...);
```

- **`s`** is the **string from which to read**
- **`format`** is the **format specifier**
- remaining arguments are as with **`scanf`**
- **returns number of conversions** or **-1 on failure**

Use `snprintf` to Write Formatted Output to a String

To write formatted output to a string:

```
int snprintf (char* s, size_t size,
       const char* format, ...);
```

- **`s`** is the **array to which to write**
- **`size`** is the **length of array `s`**
- **`format`** is the **format specifier**
- remaining arguments are as with **`printf`**
- **returns number of characters printed** or **negative number on failure**

Let's Write a Variadic Logging Function

One last topic: how to write functions
- with variable number of arguments
- (called **variadic functions**).

Say we want to **write a logging function**:
- **log output** goes **to a** specific **log file**,
- individual **calls** should **look like `printf`** (flexible, formatted output).

Call Our Logging Function `printlog`

First,
- **include C library header <stdarg.h>**
- which supports variadic functions.

Call our function **`printlog`**:

```
int printlog (const char* fmt, ...);
```

A **user might** then **write**, for example:

```
printlog ("Add %d,%d to get %d.\n",
          a, b, sum);
```

