

Use `scanf/printf` for Formatted I/O with `stdin/stdout`

You already know the shortcut versions for formatted I/O:

```
int scanf (const char* format, ...);
    reads formatted input from stdin.
```

```
int printf (const char* format, ...);
    writes formatted output to stdout.
```

Use `fscanf` to Read Formatted Input from a Stream

To read formatted input from a stream:

```
int fscanf (FILE* stream,
            const char* format, ...);
```

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

Use `fprintf` to Write Formatted Output to a Stream

To write formatted output to a stream:

```
int fprintf (FILE* stream,
             const char* format, ...);
```

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

Should Data be Stored in ASCII, or in Binary?

What's better...

data stored as ASCII

```
N_vertices=44164
START Vertex<0>
X=293540
Y=454970
Deg=2
Neighbor0=23
Neighbor1=44142
END Vertex<0>
...
```

OR

data stored
as bits

```
84 AC 00 00
A4 7A 04 00
3A F1 06 00
02 00 00 00
17 00 00 00
6E AC 00 00
```