

# **3. FORMATTED INPUT/OUTPUT**

# The `printf` Function

- The first argument in a call of `printf` is a string, which may contain both ordinary characters and *conversion specifications*, which begin with the `%` symbol.
- Ordinary characters are printed as they appear in the string; conversion specifications are matched one-by-one with the remaining arguments in the call of `printf`:

```
int i, j;  
float x, y;  
printf("i = %d, j = %d, x = %f, y = %f\n",  
       i, j, x, y);
```

# The printf Function

- Warning: C compilers don't check that the number of conversion specifications in a format string matches the number of output items:

```
printf("%d %d\n", i); /* too few items */  
printf("%d\n", i, j); /* too many items */
```

- Furthermore, there's no check that a conversion specification is appropriate for the type of item being printed:

```
printf("%f %d\n", i, x);
```