CSD101: Introduction to computing and programming (ICP)

scanf

- scanf is similar to printf in structure. It has a format specification and arguments.
- The format specification contains the conversion specifications and reads in values based on the conversion specification till it hits a white space character (blank, tab, newline, carriage return, vertical tab, formfeed). The argument corresponding to each conversion specification must be an address of a variable of a compatible type. Conversion specifications, consisting of a %, an optional assignment suppression character * , an optional number specifying a maximum field width, an optional h , l , or L indicating the width of the target, and a conversion character.
- Any characters specified in the format specification other than the conversion specification (and whitespace chars) must match exactly in the input. Except a whitespace can match one/more white space in input.
- scanf returns the number of items converted and assigned or an error (or EOF - end of file) if there is a converion error (end of input

scanf - conversion table[2]

Table B.2 *Scanf Conversions*

Character	Input Data; Argument type
`d`	decimal integer; `int*`
`i`	integer; `int*`. The integer may be in octal (leading `0`) or hexadecimal (leading `0x` or `0X`).
`o`	octal integer (with or without leading zero); `int *`.
`u`	unsigned decimal integer; `unsigned int *`.
`x`	hexadecimal integer (with or without leading `0x` or `0X`); `int*`.
`c`	characters; `char*`. The next input characters are placed in the indicated array, up to the number given by the width field; the default is 1. No `'\0'` is added. The normal skip over white space characters is suppressed in this case; to read the next non-white space character, use `%1s`.
`s`	string of non-white space characters (not quoted); `char *`, pointing to an array of characters large enough to hold the string and a terminating `'\0'` that will be added.
`e,f,g`	floating-point number; `float *`. The input format for `float`'s is an optional sign, a string of numbers possibly containing a decimal point, and an optional exponent field containing an `E` or `e` followed by a possibly signed integer.

[2] From Kernighan, Ritchie

Whitespace characters

- Whitespace characters are control characters that output some kind of empty space on the output.
- These are: horizontal tab, line feed, vertical tab, form feed, carriage return, blank/space.
- On the keyboard one can type in those characters by Ctrl-I, Ctrl-J, Ctrl-K, Ctrl-L, Ctrl-M and space-bar respectively. Small-case character after Ctrl works.
- Their ASCII codes in decimal are: 9, 10, 11, 12, 13, 32.
- In C programs they are denoted by: `\t`, `\n`, `\v`, `\f`, `\r` respectively and space is the blank character.

Other input-output functions

- The standard **C** library also provides single character input-output.
- `getchar()` reads a single character from the standard input stream (keyboard) and returns an `int` value corresponding to the character read. If the end of the input stream is reached it returns the `EOF` character (end-of-file). `EOF` is returned when there is no more input and the end of the input stream is reached.
- `putchar(ch)` writes the character ch on the standard output (terminal/screen) and returns an integer corresponding to the character ch.
- Note that **C** encodes characters as unsigned integers so one can do arithmetic operations on characters as if they were integers. This is actually a weak point of **C** .