substitutable parameters shown in *italics*

## Compilation

- gcc -flags program .c
- dcc -flags program .c (CSE labs only)
- -c Compile only, output to file .o
- -o file Executable output to file
- -g Generate debugging info for gdb
- -Wall Turn on all warnings

## Lexical structure, preprocessor

- /* comment */
- // comment to end of line
- #include <libmodule.h>
- #include "usermodule.h"
- #define NAME replacement-text
- #define NAME(args...) replacement-text

## Program structure:

*Header files:* declarations only (#includes, #defines, function prototypes)

*Implementation files:* #includes, #defines, prototypes for local functions, function definitions

*Main program file:* as for implementation, must have main:

```
int main(int argc, char **argv)
```

Identifiers start with a letter, followed by any number of letters, digits or underscores
Identifiers starting with _ reserved for system use

## Global variables and operators

**Reserved words** (can’t use as identifiers):

- auto break case char const continue default do double else entry enum extern float for goto if int long register return short signed sizeof static struct switch typedef union unsigned void volatile wait

**Reserved identifiers**:

- char prev = '\n';
- char *mssg = "hello";
- int seq[MAX] = { 1, 2, 3 };
- KeyValType keylist[] = {
  "NSW", 0, "Vic", 5, "Qld", -1 };

**Operators** (decreasing precedence down and across)

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( ) [] . - &gt;</td>
<td>Brackets, array, struct, pointer-struct</td>
</tr>
<tr>
<td>↔ ↔ - ! *</td>
<td>Inc/decrement, unary minus, logical</td>
</tr>
<tr>
<td>&amp; ~ sizeof</td>
<td>NOT, pointer deref., address-of, 1’s complement, size in bytes, cast</td>
</tr>
<tr>
<td>* / %</td>
<td>Binary arithmetic operators</td>
</tr>
<tr>
<td>&lt;&lt; &gt;&gt;</td>
<td>Bitwise left shift/right shift</td>
</tr>
<tr>
<td>&lt;= &gt;=</td>
<td>Relational operators</td>
</tr>
<tr>
<td>== != &amp;</td>
<td>(In)equality operators; bitwise AND</td>
</tr>
<tr>
<td></td>
<td>Bitwise exclusive OR, inclusive OR</td>
</tr>
<tr>
<td>&amp;&amp;</td>
<td></td>
</tr>
<tr>
<td>+= -= *= /=</td>
<td>Assignment (with optional arithmetic operation)</td>
</tr>
</tbody>
</table>

Left-associative except for **(right associative)**

## Statements

- expression ;
- { statements... }
- if (expression) statement
- if (expression) statement else statement
- switch (expression) {
  - case constant : statements...break;
  - case constant : statements...break;
  - default : statements
- }
- while (expression) statement
- for (initialiser; condition; increment) statement
- do statement while (expression);
- break; terminate loop or switch
- continue; resume next iteration of loop
- return expr; return value from function
- goto identifier; transfer to label (rare)
C library functions (and other objects)

Parameter name implies type:
c char
n int l long s string (char *)
b buffer (char array) p pointer (void *)
d double fh file handle (FILE *)

stdlib.h
atoi(s) atof(s) string to int or double
malloc(n) calloc(n) allocate n bytes
free(p) recycle memory
exit(n) terminate with status n
abs(n) labs(l) absolute value

stdio.h
fopen(s, mode)
        mode is one or more of
"r", "w", "a", "b", "+"
fclose(fh) getc(fh) getchar() close file
fgetc(b, n, fh) read char, EOF if none
fputs(c, fh) putc(char) write char
fputs(s, fh) write line
fread(p, size, n, fh) read into binary buffer,
        return number of elements read
fwrite(p, size, n, fh) write from binary buffer

Formatted output:
strftime(fh, format, list) formatted output to fh
printf(format, list) fmt output to stdout
sscanf(s, format, list) formatted output to string

Formatted input:
scanf(format, list) fmt input from stdin
sscanf(s, format, list) formatted input from string

Format codes similar to printf, list has addresses

ctype.h
toupper(c) tolower(c) case mapping
isupper(c) islower(c) case testing
isalpha(c) isalnum(c) alpha (italic) numeric
isdigit(c) isxdigit(c) decimal or hex digit
isspace(c) isprint(c) white space, printable

string.h
strlen(s) length (excluding '\0')
strncpy(ss, sd, n) copy ss to sd, return sd
strcat(ss, sd) append ss to sd, return sd
strcmp(s1, s2) compare, return <0 ==0 >0
strncpy(ss, sd, n) strncpy(ss, sd, n)
strncpy(s1, s2, n) max n chars processed
strchr(s, c) return ptr to first c in s
strchr(s, c) return ptr to last c in s
strstr(s, sp) return ptr to first sp in s
strpbrk(s, set) return ptr to first of any in set
strspn(s, set) length of prefix of any in set
strcspn(s, set) length of prefix all not in set

math.h (all parameters are double)
sin(d) cos(d) tan(d) trigonometry (radians)
asin(d) acos(d) atan(d) inverse (radians)
atan2(y, x) = tan~1(y/x)
sinh(d) cosh(d) tanh(d) hyperbolic
exp(d) log(d) log10(d) exponential, logarithm
pow(x, y) sqrt(d) x^1/2, square root
floor(d) ceil(d) integral bounds
fabs(d) fmod(x, y) absolute value, x % y